



Universidad  
Carlos III de Madrid

TRABAJO FIN DE GRADO  
Grado en Ingeniería de Sistemas Audiovisuales

# DESARROLLO DE APLICACIÓN PARA PYMES EN TECNOLOGÍA IOS

Autor: Guillermo Blanco Vera

Tutor: Carlos García Rubio

Leganés, julio de 2016



**Título:** Desarrollo de Aplicación para PYMES en Tecnología iOS

**Autor:** Guillermo Blanco Vera

**Director:** Carlos Cuadrado Ortega

## EL TRIBUNAL

**Presidente:** \_\_\_\_\_

**Vocal:** \_\_\_\_\_

**Secretario:** \_\_\_\_\_

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 5 de julio de 2016 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



# Índice

<b>1. Introduction .....</b>	<b>1</b>
1.1 Introduction of the Project .....	1
1.2 Motivation.....	2
1.3 Scope of the project .....	3
1.4 Methodology and Calendar .....	4
1.5 Work Phases .....	5
1.6 Means employed.....	6
1.7 Document Structure .....	6
<b>2. Estado del arte.....</b>	<b>7</b>
2.1 Entorno de desarrollo móvil seleccionado .....	7
2.1.1 Entornos de desarrollo móvil.....	7
2.1.2 Visión general de iOS.....	9
2.1.3 Estructura de iOS (Terminología).....	12
2.1.4 Entorno de desarrollo móvil integrado.....	14
2.2 Análisis de aplicaciones de gestión de empresas .....	14
2.2.1 Control de horas.....	15
2.2.2. Gestión de clientes .....	18
2.2.3 Comunicación.....	19
2.2.4 Gestión de recursos .....	21
2.3 Conclusiones del análisis.....	22
2.4 Marco Socio-económico .....	23
2.5 Política de privacidad y tratamiento de datos personales .....	24
<b>3 Descripción y organización.....</b>	<b>25</b>
3.1 Descripción de la aplicación a desarrollar.....	25
3.1.1 Contexto.....	25
3.1.2 Especificación de requisitos.....	27
3.1.3 Descripción de la aplicación móvil y wearable .....	28
.....	37
3.1.4 Características del usuario objetivo.....	38
3.1.5 Restricciones de diseño .....	38
3.2 Estimación del proyecto y presupuesto .....	39
3.3 Metodología de trabajo .....	41
3.3.1 SCRUM .....	41
3.3.2 Historias de usuario.....	43
<b>4 Desarrollo.....</b>	<b>55</b>
4.1 Estructura del proyecto .....	55
4.2 Funcionalidades .....	64
4.2.1 ConexionManager.....	64
4.2.2 Login.....	65
4.2.3 Módulo Menú lateral.....	65
4.2.4 Módulo Panel Principal.....	66
4.2.5 Módulo Registro de Horas .....	66
4.2.6 Módulo Correo .....	66
4.2.7 Módulo Mi perfil.....	67
4.2.8 Módulo Recursos.....	67
4.2.9 Módulo Empresas .....	68
4.2.10 APNs (Apple Push Notification service) [17] .....	68
4.2.11 GPS.....	71

4.2.12 Llamada .....	72
4.2.13 Email.....	72
4.2.14 Backend.....	73
4.3 Pruebas de calidad .....	74
<b>5. Conclusions .....</b>	<b>78</b>
5.1 Future lines .....	78
5.2 Design improvements.....	79
5.3 Code refactoring.....	79
5.4 Expansion and modules integration.....	79
<b>Bibliografía .....</b>	<b>81</b>
<b>Glosario.....</b>	<b>83</b>
<b>Anexo I: Introducción traducida al castellano .....</b>	<b>84</b>
1.1 Introducción del proyecto .....	84
1.2 Motivación .....	85
1.3 Alcance del proyecto .....	86
1.4 Metodología y calendario .....	87
1.5 Fases del trabajo .....	88
1.6 Medios empleados.....	89
1.7 Estructura del documento .....	90
<b>Anexo II: Conclusiones traducidas al castellano .....</b>	<b>91</b>
5.1 Líneas futuras.....	92
5.2 Mejora de diseño.....	92
5.3 Refactorización del código.....	92
5.4 Ampliación e integración de módulos.....	92
<b>Anexo III: Abstract .....</b>	<b>94</b>
Abstract .....	94

## Índice de Figuras

Figura 2.1.1.1 Sistemas operativos más utilizados en abril de 2016.....	7
Figura 2.1.1.2 Sistemas operativos más utilizados en abril de 2016 por versiones..	8
Figura 2.1.2.2 Versiones instaladas en dispositivos Apple .....	12
Figura 2.1.3.1 Modelo-vista-controlador .....	13
Figura 2.2.1.1.1 Capturas de pantalla del iPhone y del Apple Watch de la aplicación Harvest.....	15
Figura 2.2.1.2.1 Capturas de pantalla del iPhone de la aplicación Clockson.....	16
Figura 2.2.1.3.1 Capturas de pantalla del iPhone de la aplicación TimeSheet.....	17
Figura 2.2.2.1.1 Capturas de pantalla del iPhone de la aplicación SugarCRM.....	18
Figura 2.2.3.1.1 Capturas de pantalla del iPhone de la aplicación Noysi.....	19
Figura 2.2.3.2.1 Capturas de pantalla del iPhone de la aplicación Mmmelon.....	20
Figura 2.2.4.1.1 Capturas de pantalla del iPhone de la aplicación MiniKeePass.....	21
Figura 2.3.1 Número de clientes de telefonía móvil en febrero de 2016.....	23
Figura 3.1.1.1 Aplicación web de Armadillo Amarillo.....	25
Figura 3.1.1.2 Módulos del <i>backend</i> .....	26
Figura 3.1.1.3 Tablas de la base de datos.....	26
Figura 3.1.3.2 Pantalla de la aplicación móvil del módulo de Login.....	28
Figura 3.1.3.2 Pantalla de la aplicación móvil del módulo Menú Lateral.....	29
Figura 3.1.3.3 Pantallas de la aplicación móvil del módulo Panel Principal.....	30
Figura 3.1.3.4 Pantallas de la aplicación móvil del módulo Registro de Horas.....	31
Figura 3.1.3.5 Pantallas de la aplicación móvil del módulo Correo.....	32
Figura 3.1.3.6 Pantalla de la aplicación móvil del módulo Mi Perfil.....	33
Figura 3.1.3.7 Pantallas de la aplicación móvil del módulo Ajustes de Mi Perfil.....	34
Figura 3.1.3.8 Pantallas de la aplicación móvil del módulo Recursos.....	35
Figura 3.1.3.9 Pantallas de la aplicación móvil del módulo Empresas.....	36
Figura 3.1.3.10 Pantallas de la aplicación móvil del módulo Cómo Llegar de Empresas.....	37
Figura 3.1.3.11 Pantallas de la aplicación <i>wearable</i> .....	37
Figura 3.3.1.1 Ejemplo de Historias de Usuario en Jira.....	42
Figura 3.3.1.2 Ejemplo de un proyecto en SourceTree.....	42
Figura 4.1.1 Organización de las carpetas del proyecto.....	55
Figura 4.1.2 Organización de las carpetas de la aplicación móvil.....	56
Figura 4.1.3 Diferentes tamaños de imágenes para diferentes dispositivos.....	56
Figura 4.1.4 Organización de los Storyboards de la aplicación móvil.....	57
Figura 4.1.5 Ejemplo de Storyboard de la aplicación móvil.....	57
Figura 4.1.6 Organización de la carpeta de modelos de la aplicación móvil.....	58
Figura 4.1.7 Ejemplo de modelo <i>User</i> . Campos de la tabla <i>user</i> de la base de datos.....	58
Figura 4.1.8 Organización de la carpeta <i>Helpers</i> de la aplicación móvil.....	59
Figura 4.1.9 Organización de carpeta Managers de la aplicación móvil.....	59
Figura 4.1.10 Organización de la carpeta <i>VlewComtrollers</i> de la aplicación móvil.....	60
Figura 4.1.11 Carpeta Login.....	60
Figura 4.1.12 Carpeta SWReveal.....	60
Figura 4.1.13 Carpeta <i>Principal Panel</i> .....	61
Figura 4.1.14 Carpeta <i>Time Track</i> .....	61
Figura 4.1.15 Carpeta <i>Mail</i> .....	62

Figura 4.1.16 Carpeta <i>My Profile</i> .....	62
Figura 4.1.17 Carpeta <i>Resources</i> .....	63
Figura 4.1.18 Carpeta <i>Companies</i> . ....	63
Figura 4.1.19 Carpeta <i>Supporting Files</i> . ....	64
Figura 4.2.10.1 Trayecto de una notificación push.....	69
Figura 4.2.10.2 Esquema de autenticación TLS.....	70
Figura 4.2.10.3 Esquema de generación de token.....	70
Figura 4.2.10.3 Esquema de envío de <i>payload</i> .....	71



## Índice de Tablas

Tabla 2.1.2.1 Historial de versiones de iOS [5] .....	11
Tabla 3.2.1 Estimaciones en horas del proyecto.....	39
Tabla 3.2.2 Costes de Personal .....	40
Tabla 3.3.3 Costes de material .....	40
Tabla 3.3.3 Presupuesto total.....	40
Tabla 3.3.2.1.1 Historia de usuario Diseño Login.....	43
Tabla 3.3.2.1.2 Historia de usuario Hacer Login.....	43
Tabla 3.3.2.1.3 Historia de usuario Gestión de errores Login.....	43
Tabla 3.3.2.1.4 Historia de usuario Diseño Menú Lateral.....	43
Tabla 3.3.2.1.5 Historia de usuario Funcionalidad Menú Lateral.....	43
Tabla 3.3.2.1.6 Historia de usuario Diseño Panel Principal.....	44
Tabla 3.3.2.1.7 Historia de usuario Funcionalidad Panel Principal.....	44
Tabla 3.3.2.1.8 Historia de usuario Diseño Registro de Horas.....	44
Tabla 3.3.2.1.9 Historia de usuario Funcionalidad Registro de Horas.....	45
Tabla 3.3.2.1.10 Historia de usuario Diseño Añadir Imputación.....	45
Tabla 3.3.2.1.11 Historia de usuario Modificación peticiones backend.....	45
Tabla 3.3.2.1.12 Historia de usuario Funcionalidad MVC .....	45
Tabla 3.3.2.2.1 Historia de usuario Funcionalidad Editar/Añadir Imputación.....	46
Tabla 3.3.2.2.2 Historia de usuario Diseño Recursos.....	46
Tabla 3.3.2.2.3 Historia de usuario Funcionalidad Recursos.....	46
Tabla 3.3.2.2.4 Historia de usuario Diseño Crear/Ver/Editar Recurso.....	47
Tabla 3.3.2.2.5 Historia de usuario Funcionalidad Crear/Ver/Editar Recurso.....	47
Tabla 3.3.2.2.6 Historia de usuario Diseño Mi Perfil.....	47
Tabla 3.3.2.2.7 Historia de usuario Funcionalidad Mi Perfil.....	47
Tabla 3.3.2.2.8 Historia de usuario Diseño Editar Perfil.....	48
Tabla 3.3.2.2.9 Historia de usuario Funcionalidad Editar Perfil.....	48
Tabla 3.3.2.2.10 Historia de usuario Diseño configurar Notificaciones.....	48
Tabla 3.3.2.2.11 Historia de usuario Funcionalidad Configuración Notificaciones.....	48
Tabla 3.3.2.2.12 Historia de usuario Diseño Acerca de.....	49
Tabla 3.3.2.2.13 Historia de usuario Funcionalidad Acerca de.....	49
Tabla 3.3.2.3.1 Historia de usuario Diseño Mensajes.....	49
Tabla 3.3.2.3.2 Historia de usuario Funcionalidad Mensajes.....	50
Tabla 3.3.2.3.3 Historia de usuario Diseño Vista Mensajes.....	50
Tabla 3.3.2.3.4 Historia de usuario Funcionalidad Vista Mensaje.....	50
Tabla 3.3.2.3.5 Historia de usuario Diseño Nuevo/Responder Mensaje.....	51
Tabla 3.3.2.3.6 Historia de usuario Funcionalidad Nuevo/Responder Mensaje.....	51
Tabla 3.3.2.3.7 Historia de usuario Diseño Cómo llegar.....	51
Tabla 3.3.2.3.8 Historia de usuario Funcionalidad Cómo llegar.....	51
Tabla 3.3.2.3.9 Historia de usuario Que los recursos aparezcan en función de los usuarios.....	52
Tabla 3.3.2.3.10 Historia de usuario Traducir toda la app.....	52
Tabla 3.3.2.3.11 Historia de usuario Meter versión de la app en “About”.....	52
Tabla 3.3.2.3.12 Historia de usuario La funcionalidad “Set Date” que no aparezca si vienes del calendario.....	52
Tabla 3.3.2.3.13 Historia de usuario Cambiar foto de mi perfil.....	52
Tabla 3.3.2.4.1 Historia de usuario Notificaciones Backend.....	52
Tabla 3.3.2.4.2 Historia de usuario Funcionalidad Configuración Notificaciones.....	53
Tabla 3.3.2.4.3 Historia de usuario Diseño de la vista del wearable.....	53

Tabla 3.3.2.4.4 Historia de usuario Funcionalidad wearable.....	53
Tabla 3.3.2.4.5 Historia de usuario Modificación petición mensajes Backend.....	53
Tabla 3.3.2.4.6 Historia de usuario Notificaciones Wearable Backend. ....	54
Tabla 3.3.2.5.1 Historia de usuario Hacer capturas de pantalla de las dos plataformas. ....	54
Tabla 3.3.2.6.1 Historia de usuario Memoria iOS. ....	54
Tabla 3.3.2.6.2 Historia de usuario Bugs iOS.....	54
Tabla 3.3.2.6.3 Historia de usuario Diseño iOS.....	54
Tabla 4.2.14.1 Funciones implementadas en el <i>backend</i> . ....	73

# 1. Introduction

## 1.1 Introduction of the Project

Nowadays, the telecommunication world and, specifically, the one of the mobile telephony is constantly increasing and the technical innovations commonly used in the current world

The use of mobile phones and the purchase of smartphones is more habitual in more sectors of the population and there appear new needs that are covered by this devices.

That's the reason why I started to carry out my practices of the University in a company called *Armadillo Amarillo S.C.*<sup>1</sup>, dedicated to the development of web and mobile applications. During this time, I learned how to use the programming language *Objective-C* of *Apple*. At the beginning, I used to develop easy programs in order to familiarize with the environment and with this language, but in a few days I started to take care of certain tasks of real projects.

That was when I was asked to do the Final Year with them, a proposal that I accepted without any doubt. The work was to be an application for SMEs called INTRANET.

The development would take place in native language, that is, in the aforementioned language of *Apple*, *Objective-C*, and it was to be the extension of an existing project in the company. This project consisted of a web application performing a series of tasks of internal control of the company and employees and a prototype mobile application. The idea was to create two applications for the two most common operating systems on the market: iOS and Android. Logically, I would be responsible for the iOS part and a partner would be responsible for the Android part.

Both applications have the same modules: one to perform the time track that every employee expend in each project, called **Time Track**; another one to send and receive messages between the members of a company, denominated **Mail**; the third module will be responsible for showing the user profile and for allowing changes in personal information, **My Profile**; the module **Resources** will show the resources an user have access to; and, finally, the last module is the one

---

<sup>1</sup> Armadillo Amarillo S.C., 2016, [online]. Available in: <http://www.armadilloamarillo.com/> [consulted: 1 de junio de 2016]

that list the client companies, their contact people and gives the option of opening a maps application in order to show the route to this company, called **Companies**.

We did the design of both applications. Both my partner and I (each our own application) take care of outline the different screens that will be shown so that, lately, we could develop them. Always taking care that both applications were as similar as possible.

The applications bear a part in the same *backend* and the same database. The *backend* existed, since it was the one used by the web application, nevertheless we had to enlarge it and decided to split, by modules, the work of implementing the different methods that will be necessary so that both applications work correctly.

By last, the project will incorporate a *wearable* application. This application will manage the push notifications.

## 1.2 Motivation

Nowadays, more and more people get a smartphone. It is why the world of mobile applications is booming and the trend towards its development is growing.

It is no wonder that the companies are interested in digitizing the most of the daily work of the offices and the tendency to use mobile terminals for most management is increasing.

Because of that mi company offers me to perform this project. INTRANET is an application that meets the main daily needs of the employees. If we have a look at the market, we can find some apps which functionalities are similar to some of the modules mentioned in the preceding paragraph, but none has all. So, this project can be useful for many companies.

Personally, I consider myself a person who tries to be informed of the technical developments emerging in various technological fields but, specifically, I feel great interest in innovations in mobile technologies. That is why I usually try to acquire the last brought out terminal, download new applications to test them, see how far they can go and find out bugs.

In addition, I usually have ideas to create applications that I couldn't get to develop before. It is why I was interested in do my TFG of a mobile application in a company. Is a way of thanking *Armadillo Amarillo S.C.* what they have taught me and how they gave me some autonomy and the possibility of shaping my ideas.

Finally, six years ago I bought an iPhone 4 and a year later I purchased my first MacBook Pro. Since then, I have felt great admiration for Apple and learn how to develop applications and to be able to test them my own was one of the goals I raised since I studied different programming subjects that were proposed in my degree.

### 1.3 Scope of the project

The project can be divided into three main categories: mobile application, wearable application and *backend*.

- **Mobile application**

- Login: Screen that appears when the app starts in order to introduce the username and the password.
- Principal Panel: Screen that appears when login. It shows a sum up of the last three time tracks, the last three mails and the last three resources.
- Time Track: module of the application where the user can impute the hours it have dedicated to a particular project and take advice of the time track of a specific day.
- Mail: module of the application that consist of a mail system in order to send and receive messages between the employees of a company.
- My Profile: screen in which the user can take advice and modify its profile or activate or deactivate the notifications.
- Resources: application module that consist of consult certain usernames and passwords that can be useful for some employees.
- Companies: application module in which the user can see the customers of the company, its contact people and launch an external application with the location of this customer's office.
- Lateral Menu: this is a side menu from which the user can access all modules and views mentioned.

- **Wearable application**

- Notifications: the Apple Watch will show a notification when the user receives a new message and allows it to answer with a series of pre-set messages.

- **Backend**

- Services: through backend services we get a communication between the application and the database.
- APNS (Apple Push Notification Service): push notifications are handled through the backend with calls to Apple Push Notification Service.

## 1.4 Methodology and Calendar

For the project development I followed the methodology conducted by my company, that is a mixture between the Scrum methodology and the distribution of tasks in *sprints*.

I used a version driver called SourceTree<sup>2</sup> that save every change at runtime and allows going back to any previous state of the project. These changes are instantly stored locally and remotely, so the user can access its project from anywhere with the only requirement to have an Internet connection.

On the other hand, it carried out a division of labour in *sprints*. A *sprint* is a set of specific tasks of the project called user stories. The team meets after finishing each *sprint*, that are two weeks long in our case, in order to see the progress and possible problems that may have arisen.

The schedule we followed was as follows:

- Sprint 0 from February 1 until February 9, 2016.
- Sprint 1 from February 10 until February 24, 2016.
- Sprint 2 from February 25 until March 20, 2016.
- Sprint 3 from March 20 until March 27, 2016.
- Sprint 4 from March 28 until April 11, 2016.
- Sprint 5 from April 12 until April 26, 2016.
- Sprint 6 from April 27 until May 11, 2016.

---

<sup>2</sup> Driver versions SourceTree: <https://www.sourcetreeapp.com/>

## 1.5 Work Phases

The Project has been divided into three phases: conceptualization, design and analysis; development and QA (Quality Assurance or quality testing); and writing memory.

- **Conceptualization, design and analysis**
  - Sprint 0
    - Conceptualize the idea of the application.
    - Designing all views of the application.
- **Development and QA**
  - Sprint 1
    - Design and functionality of Login/Logout.
    - Design and functionality of Lateral Menu.
    - Design of the Principal Panel.
    - Implementation of the Connection Manager.
  - Sprint 2
    - Design and functionality of Time Track.
    - Design and functionality of Resources.
    - Design and functionality of My Profile.
  - Sprint 3
    - Functionality of Principal Panel.
    - Design and functionality of Mail.
  - Sprint 4
    - Design and functionality of Companies.
    - Integration with APNS (Apple Push Notification Service).
  - Sprint 5
    - Design and functionality of Edit Profile Photo.
    - Notification Management.
    - Design and functionality of Apple Watch.
- **Memory writing**
  - Sprint 6
    - Writing memory

## 1.6 Means employed

For the development of this project I used the following means:

- **Hardware:**
  - MacBook 8Gb RAM and SSD 250Gb
  - iMac 8Gb RAM and SSD 250Gb
  - iPhone 5
  - iPhone 6s
  - Apple Watch 38mm
  
- **Software**
  - OS X El Capitan (v. 10.11.3)
  - iOS 8.4
  - iOS 9.3.1
  - watchOS 2

## 1.7 Document Structure

I have divided memory into 5 chapters: introduction, state of the art, description and organization, development and conclusions.

In the chapter of the state of the art it is made an analysis of the development environment that has been used, it is performed a study of similar applications that are on the market and the agile methodologies are evaluated.

In the third chapter, description and organization, first the project status is explained at the time the development started. It also shows how the development process has been divided and what has been done in each part, in addition to explaining the methodology that has been followed and the estimates that were made.

In the chapter of development it is explained, module by module, how each feature is scheduled, broadly explained. Certain design alternatives are also studied and it is shown how the project has been divided into directories for an easy intelligibility in order to have everything tidy. At the end of the chapter, the tests that have submitted the application to identify errors or bugs in the programming are explained.

The last chapter is the one of the conclusions. It explains how the application might evolve in the future, with certain modifications or expansion and /or integration of new modules.



## 2. Estado del arte

En este capítulo realizaré un análisis de los entornos de desarrollo móvil que existen y explicaré por qué he escogido iOS. Además, mostraré alguna aplicación existente cuyas funcionalidades son similares a las de la INTRANET. También hay un apartado en el que hablaré de las políticas de privacidad y tratamiento de datos ya que la aplicación obtiene información personal del usuario final. Y, para finalizar, expondré las conclusiones del análisis.

### 2.1 Entorno de desarrollo móvil seleccionado

En este apartado se realiza un análisis de los entornos de desarrollo más utilizados actualmente y, en concreto, del entorno de desarrollo que he elegido: iOS.

#### 2.1.1 Entornos de desarrollo móvil

En la actualidad existen varios entornos de desarrollo móvil, tales como: Android, iOS, Windows Phone y otros históricos, como Java ME, BlackBerry o Symbian.

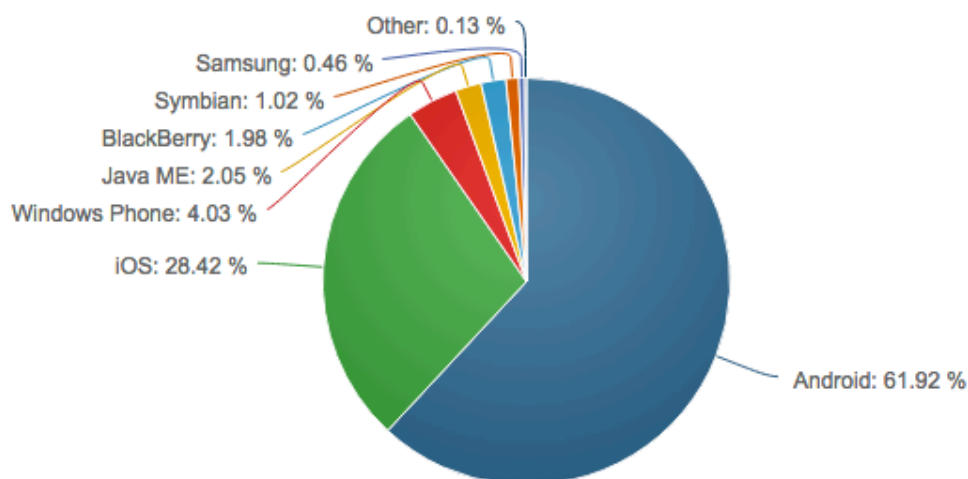


Figura 2.1.1.1 Sistemas operativos más utilizados en abril de 2016.<sup>3</sup>

<sup>3</sup> Netmarketshare, "Mobile/Tablet Operating System Market Share", 2016, [en línea]. Disponible en: <https://www.netmarketshare.com> [consulta: 2 de junio de 2016]

Podemos observar en la figura que Android es el sistema operativo más usado en la actualidad, seguido por iOS. Android es utilizado en un gran número de terminales de diferentes marcas, como por ejemplo Samsung, HTC, Huawei o BQ, entre otros, a diferencia de iOS que es instalado sólo en terminales iPhone y iPad.

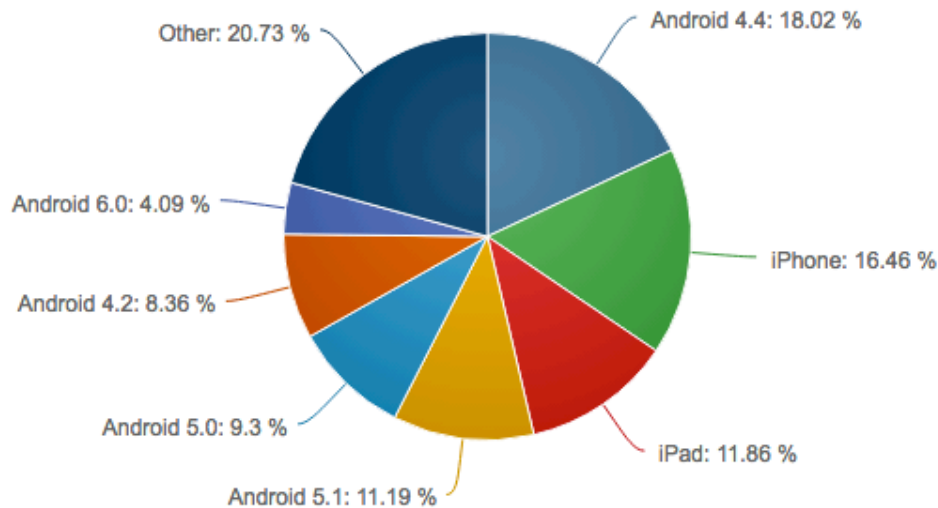


Figura 2.1.1.2 Sistemas operativos más utilizados en abril de 2016 por versiones<sup>4</sup>.

Android es un sistema operativo mucho más fragmentado que iOS, lo que trae quebraderos de cabeza tanto a usuarios como a desarrolladores, ya que tienen que contar con un gran número de terminales diferentes a la hora de programar una aplicación, sobre todo cuando realizan el diseño, obligados a tener en cuenta los diferentes formatos de los diferentes terminales que utilizan Android.

Además, la telefonía móvil también sigue una moda y unas tendencias concretas. Las aplicaciones tienen que adaptarse a las necesidades actuales de los usuarios y, por ello, podemos observar las siguientes 7 tendencias [3]:

- **Servicios Cloud:** es fundamental que el usuario tenga la posibilidad de almacenar datos en la nube sin necesidad de ocupar espacio de almacenamiento en su propio terminal y que tenga acceso a esa información desde cualquier lugar y desde cualquier dispositivo.
- **Seguridad:** muy ligada con el punto anterior, esta tendencia es cada día más necesaria porque cada vez es más frecuente compartir datos personales con los servicios cloud y por ello se debe de prestar especial atención en que una aplicación sea segura para el usuario.

<sup>4</sup> Netmarketshare, "Mobile/Tablet Operating System Market Share", 2016, [en línea]. Disponible en: <https://www.netmarketshare.com> [consulta: 2 de junio de 2016]

- **Localización:** es fundamental para aquellas aplicaciones que precisen conocer la ubicación exacta del usuario de cara a poder ofrecerle información de su entorno.
- **Wearables:** cada vez es menos raro ver a alguien por la calle que esté tocando la pantalla de su reloj. Las marcas están invirtiendo en sacar a la venta diferentes modelos de smartwatches. Es, sin duda, la mayor tendencia.
- **Marketing de aplicaciones:** las aplicaciones no se venden solas y existe una gran competencia, lo que ha creado la necesidad de diferenciarse del resto. Esto se consigue con fuertes campañas de publicidad y marketing para dar a conocer las nuevas apps.
- **Internet de las cosas:** aparentemente ciencia ficción, no es extraño que estemos en el salón y cambiemos la temperatura del horno desde el sofá. El internet de las cosas (IoT, Internet of Things) está creando aparatos conectados a plataformas cloud que recopilan información constantemente y nos facilitan labores cotidianas, dotando de inteligencia a nuestras casas, oficinas, etc.
- **Experiencia de usuario:** la más importante y quizá la más antigua de las tendencias es la experiencia de usuario. No nos podemos olvidar de que, además de la funcionalidad, el hecho de que una aplicación sea bonita e intuitiva es lo que busca el usuario final.

### 2.1.2 Visión general de iOS

Siempre me ha gustado la forma de pensar de Apple y quería comprobar si desde la programación también me convencería. Y lo ha conseguido con creces.

iOS es el sistema operativo móvil creado por *Apple Inc.* que se desarrolló en un inicio para el iPhone y que actualmente es instalado en otros dos dispositivos: el iPod Touch y el iPad. Fue anunciado por la compañía en la *Macworld Conference & Expo* el 9 de enero de 2007 con el nombre de iPhone OS, nombre que siguió llevando hasta el 7 de junio de 2010 cuando, junto al lanzamiento del iPhone 4, Steve Jobs desveló su nueva identidad: iOS [4].

El historial de versiones de iOS se muestra a continuación, junto con la fecha de lanzamiento, los dispositivos que las soportan y ciertos comentarios:

<b>Versión</b>	<b>Fecha de lanzamiento</b>	<b>Dispositivos que lo soportan</b>	<b>Comentarios</b>
iPhone OS	29 de junio de 2007	iPhone y iPod Touch (1ª generación)	No tenía nombre específico. Se decía que el iPhone corría OS X, sistema operativo de los ordenadores Mac.
iPhone OS 2.0	11 de julio de 2008	iPhone, iPhone 3G y iPod Touch (1ª y 2ª generación)	Los dispositivos que corrían la versión 1.x podían actualizar. Aparición de App Store, que permitía descargar aplicaciones de terceros.
iPhone OS 3.0	17 de junio de 2009	iPhone, iPhone 3G, iPhone 3GS, iPod Touch (1ª, 2ª y 3ª generación) y iPad (1ª generación)	Los dispositivos que corrían la versión 2.x podían actualizar.
iOS 4: Todo vuelve a cambiar, otra vez	21 de junio de 2010	iPhone 3G, iPhone 3GS, iPhone 4, iPod Touch (2ª, 3ª y 4ª generación) y iPad (1ª y 2ª generación)	Primera versión que no permite actualizar a dispositivos antiguos. Deja de cobrarse la actualización para iPod Touch.
iOS 5: Llevamos iOS a un nivel totalmente nuevo	6 de junio de 2011	iPhone 3GS, iPhone 4, iPhone 4s, iPod Touch (3ª y 4ª generación) y iPad (1ª, 2ª y 3ª generación)	No permite actualizar a dispositivos antiguos.
iOS 6: El sistema operativo móvil más avanzado de la historia	12 de junio de 2012	iPhone 3GS, iPhone 4, iPhone 4s, iPhone 5, iPod Touch (4ª y 5ª generación), iPad (2ª, 3ª y 4ª generación) y iPad Mini	No permite actualizar a dispositivos antiguos.

iOS 7: Una nueva perspectiva para el sistema operativo móvil	10 de junio de 2013	iPhone 4, iPhone 4s, iPhone 5, iPhone 5s, iPhone 5c, iPod Touch (5ª generación), iPad (2ª, 3ª y 4ª generación) y iPad Mini	No permite actualizar a dispositivos antiguos. Cambia por completo el diseño del sistema.
iOS 8: El mayor lanzamiento de iOS hasta hoy	2 de junio de 2014	iPhone 4s, iPhone 5, iPhone 5s, iPhone 5c, iPhone 6, iPhone 6 Plus, iPod Touch (5ª generación), iPad (2ª, 3ª, 4ª y 5ª generación) y iPad Mini (1ª y 2ª generación)	No permite actualizar a dispositivos antiguos.
iOS 9: El epicentro del cambio	16 de septiembre de 2015	iPhone 4s, iPhone 5, iPhone 5s, iPhone 5c, iPhone 6, iPhone 6 Plus, iPhone 6s, iPhone 6s Plus, iPhone SE, iPod Touch (5ª y 6ª generación), iPad (2ª, 3ª, 4ª y 5ª generación), iPad Mini (1ª, 2ª y 3ª y 4ª generación), iPad Air (1ª y 2ª generación) y iPad Pro	No permite actualizar a dispositivos antiguos.

Tabla 2.1.2.1 Historial de versiones de iOS [5]

Los dispositivos que están utilizando ahora mismo las diferentes versiones de iOS se pueden ver en la siguiente gráfica:

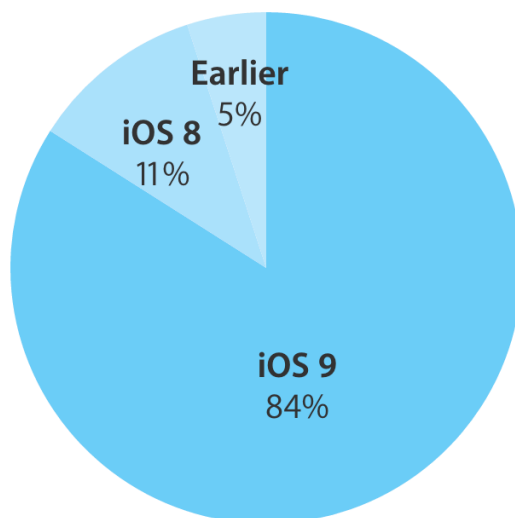


Figura 2.1.2.2 Versiones instaladas en dispositivos Apple <sup>5</sup>.

Observamos que actualmente el 84% de los usuarios con dispositivos Apple tienen instalado iOS 9, que tan sólo un 11% mantiene iOS 8 y un 5% tiene versiones más antiguas. Esto puede deberse a que sus dispositivos no dejen actualizar a nuevas versiones de iOS. Por ello también decidí que la INTRANET soportase las dos versiones mayoritarias: iOS 8 y iOS 9.

### 2.1.3 Estructura de iOS (Terminología)

iOS es un sistema operativo que utiliza el lenguaje de programación orientado a objetos *Objective-C*, que podría entenderse como un superconjunto de C. Apple ha lanzado un nuevo lenguaje de programación llamado *Swift*, que ya va por su segunda versión, pero yo utilizaré el primero de los dos mencionados para el desarrollo de la app.

---

<sup>5</sup> Apple, "Apple Store", *Apple Developer*, 2016, [en línea]. Recuperado de: <https://developer.apple.com/support/app-store/> [consulta: 3 de junio de 2016]

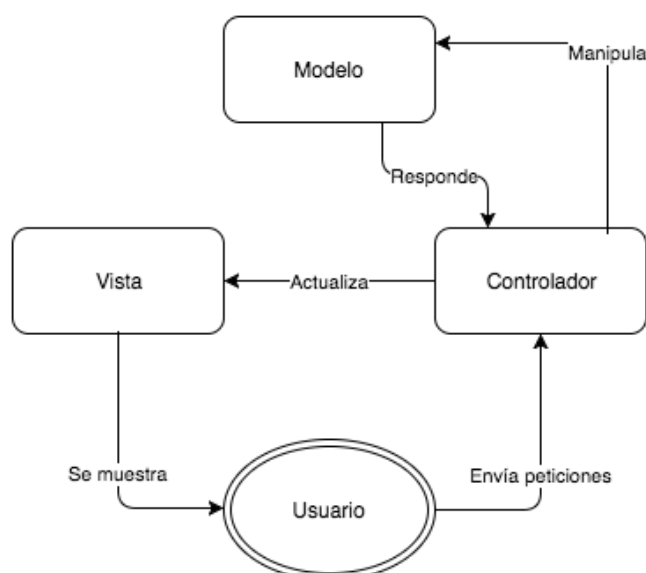


Figura 2.1.3.1 Modelo-vista-controlador <sup>6</sup>

El código basa su estructura en el **modelo-vista-controlador (MVC)**. Este patrón de arquitectura de software defiende la separación de conceptos y la reutilización de código para que la aplicación sea más eficiente y más limpia, con el objetivo de facilitar el desarrollo y el mantenimiento. El modelo contiene la información que se desea mostrar en cada momento y para ello utiliza al controlador. El controlador, por su parte, se encarga de realizar las peticiones que el modelo le solicita y enviar ciertos comandos a su vista asociada. Por último, la vista presenta el modelo de tal forma que el usuario lo entienda y pueda interactuar con él.

En cuanto al proyecto en sí, se divide en varias partes. Por un lado están las clases o **ViewControllers**. Estos son los encargados de la lógica y la interfaz de la aplicación y siguen la estructura de C, es decir, cada uno tiene dos archivos, uno con la extensión “.h”, que es la parte pública de la clase, y otro con la extensión “.m”, que es la parte privada y contiene toda la lógica.

Existe una clase especial, llamada **AppDelegate**, que contiene las clases que se encargan de gestionar el comportamiento básico de la aplicación, es decir, qué ha de hacer la app cuando se arranca, se cierra o cuando pasa a estar en segundo plano y la volvemos a abrir.

Otra parte del proyecto son los **Storyboards**. Son los encargados de la interfaz de la aplicación y podemos mostrarlos con el *interface builder* de Xcode. Nos permiten introducir en las vistas todo tipo de elementos gráficos como botones, listas, labels, etc. Además, con ellos podemos gestionar la navegación entre las diferentes ventanas de nuestra aplicación. Por último, los Storyboards tienen un elemento muy útil, denominado *constraints*, que se utiliza para adaptar

<sup>6</sup> Figura realizada en la página web: <https://www.draw.io>

las interfaces a cualquier dispositivo, independientemente de cual sea su tamaño y orientación.

Para terminar, existe una carpeta que contiene todas las imágenes de la aplicación, llamada **Images.xcassets**. En ella se introducen los diferentes tamaños de las imágenes para que se adapten automáticamente a la resolución del dispositivo que está lanzando la app.

#### 2.1.4 Entorno de desarrollo móvil integrado

Para desarrollar una aplicación iOS es necesario utilizar el entorno de desarrollo **Xcode**<sup>7</sup>, programa que sólo funciona en ordenadores Mac.

Para simular la aplicación e irla probando mientras realizas el desarrollo, Xcode te proporciona un simulador, que puede ser configurado con cualquier modelo de dispositivo Apple, ya sea iPhone, iPad, Apple Watch, etc. También puedes instalar el prototipo de la aplicación en un dispositivo real, pero para ello necesitas una cuenta de desarrollador de Apple. Deberás, además, registrar la aplicación en tu cuenta y los dispositivos en los cuales quieres instalarla y, por último, generar un certificado con esos datos para que sea posible esa prueba en dicho dispositivo.

#### 2.2 Análisis de aplicaciones de gestión de empresas

Es necesario hacer un análisis de mercado para comprobar qué aplicaciones existen similares a la INTRANET en el mercado. La aplicación tiene como objetivo la gestión de empresas, por ello he realizado una búsqueda de apps relacionadas con ese ámbito y he encontrado las siguientes:

- Control de horas:
  - Harvest
  - Clockson
  - TimeSheet
- Gestión de clientes:
  - SugarCRM Mobile
- Comunicación:
  - Noysi
  - Mmmelon
- Gestión de recursos:
  - MiniKeepass

---

<sup>7</sup> Apple, "What's new in Xcode 7.", *Apple Developer*, Xcode, 2016, [en línea]. Disponible en: <https://developer.apple.com/xcode> [consulta: 3 de junio de 2016]



## 2.2.1 Control de horas



### 2.2.1.1 Harvest [6]

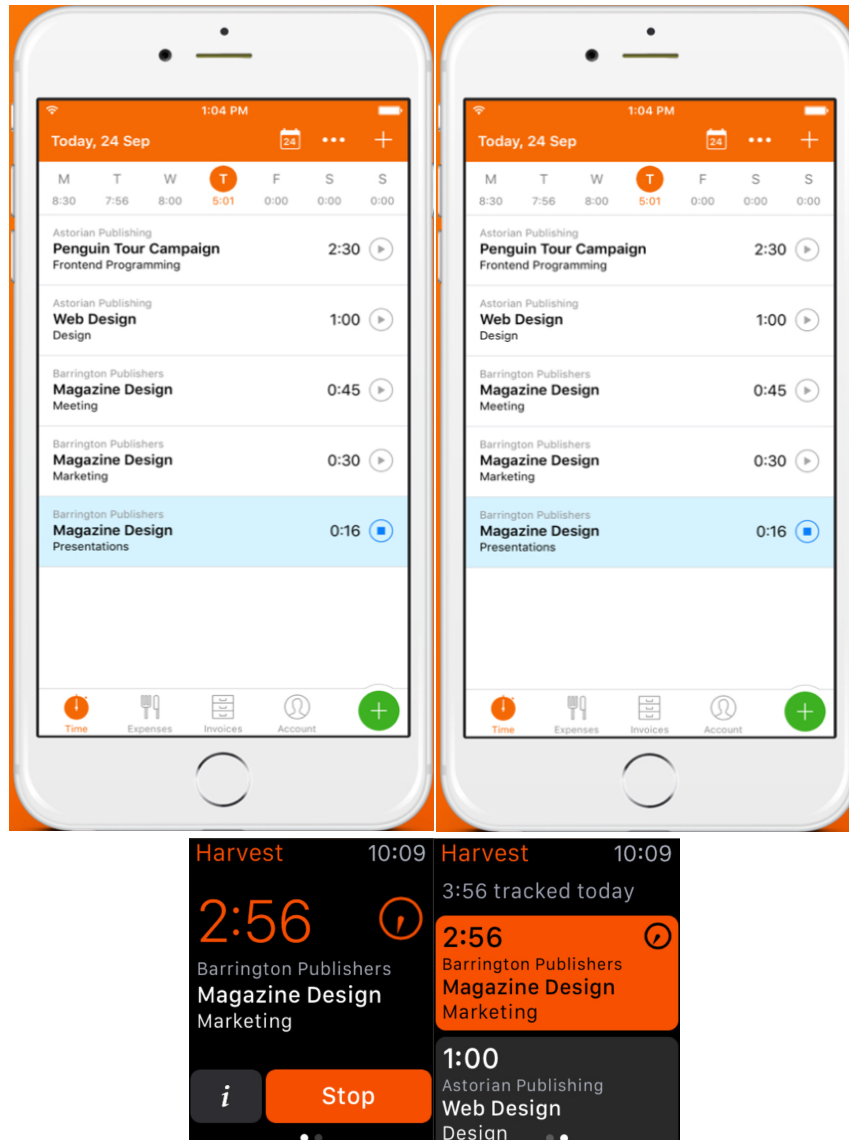


Figura 2.2.1.1.1 Capturas de pantalla del iPhone y del Apple Watch de la aplicación Harvest<sup>8</sup>.

Harvest, entre otras funcionalidades, te permite gestionar tus horas de trabajo. Cuenta con un cronómetro que puedes iniciar y parar cuando le dediques tiempo a un proyecto determinado. De esta forma podrás manejar mejor el tiempo que empleas en cada tarea. También puedes introducir las horas de manera manual. Además, te permite consultar registros de horas de otros días.

Incluye una aplicación para el Apple Watch, que te notifica el tiempo que llevas dedicado a cada proyecto, entre otras funcionalidades.

<sup>8</sup> iTunes, Harvest. Capturas de pantalla del iPhone y del Apple Watch. [Recuperadas de: <https://itunes.apple.com/es/app/harvest-time-expense-tracker/id355395846?mt=8>] [consulta: 3 de junio de 2016]



### 2.2.1.2 Clockson [7]

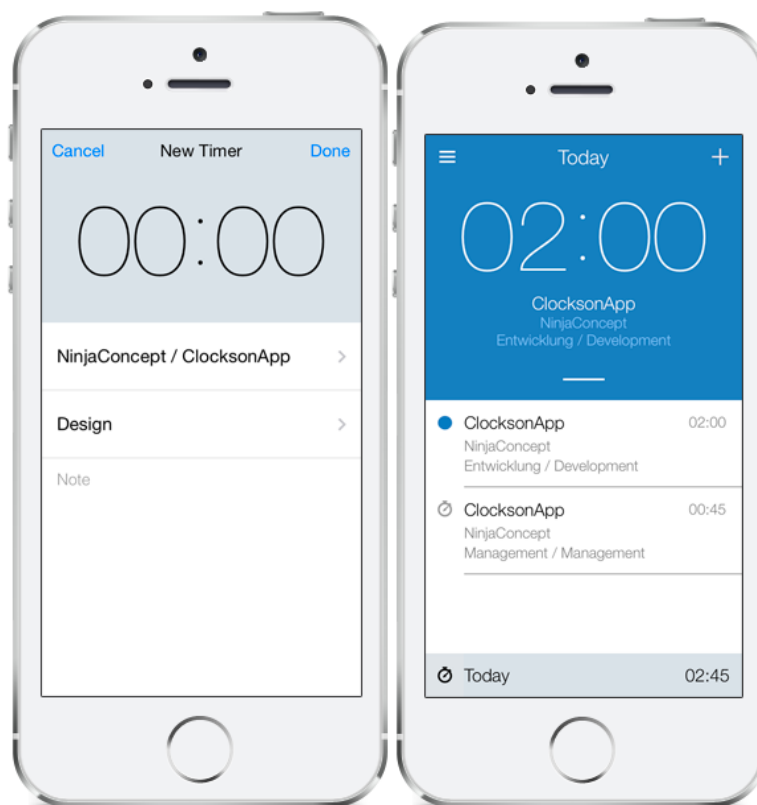


Figura 2.2.1.2.1 Capturas de pantalla del iPhone de la aplicación Clockson<sup>9</sup>.

Clockson es una aplicación que te ayuda a llevar un control de las horas que le dedicas a cada proyecto exclusivamente. Es muy intuitiva y fácil de usar.

Cuenta con un cronómetro que puedes iniciar y parar cuando lo desees. De esta forma, si haces una parada, puedes pausar el cronómetro con el objetivo de tener un control más exacto. También te permite modificar un registro y ver los registros de días pasados.

<sup>9</sup> iTunes, Clockson. Capturas de pantalla del iPhone. [Recuperadas de: <https://itunes.apple.com/es/app/clockson/id872703970?mt=8>] [consulta: 3 de junio de 2016]



### 2.2.1.3 TimeSheet [8]

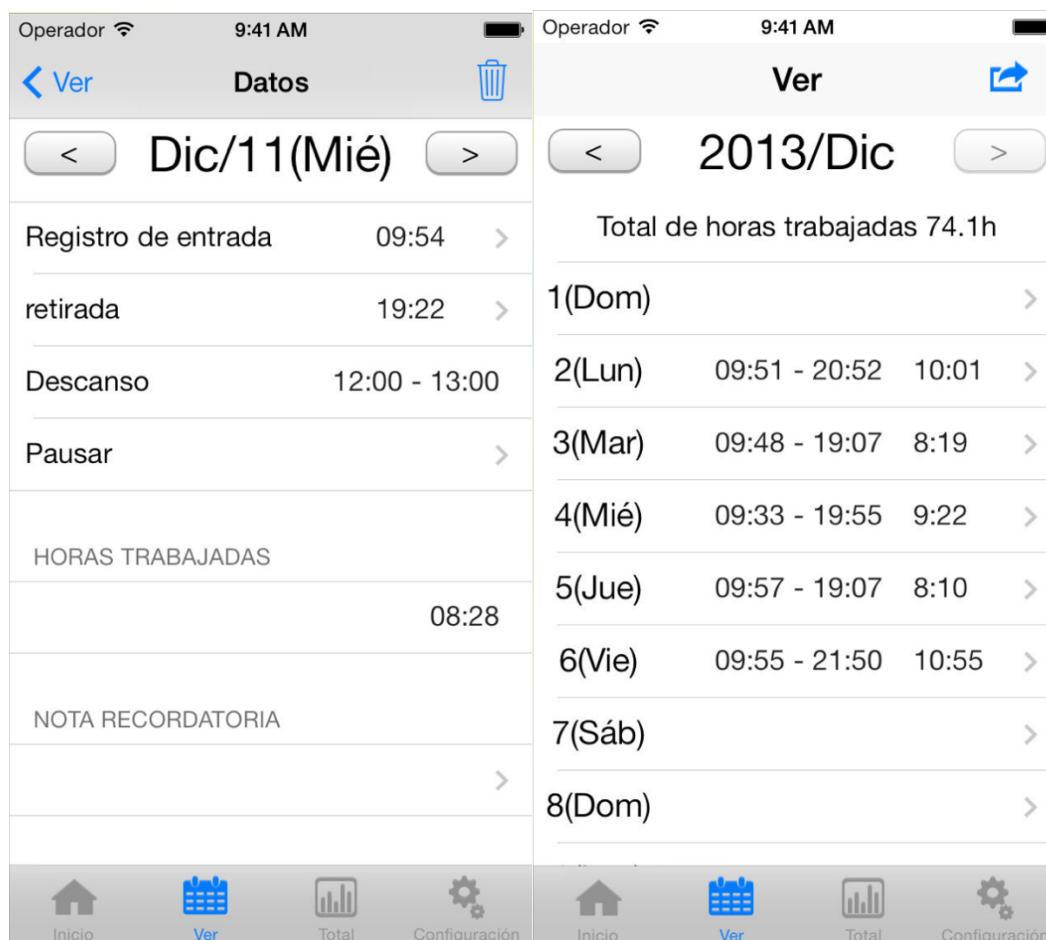


Figura 2.2.1.3.1 Capturas de pantalla del iPhone de la aplicación TimeSheet<sup>10</sup>

TimeSheet es una herramienta que te ayuda a realizar un seguimiento de las horas laborales. Muestra resúmenes por días o meses, haciendo hincapié en las horas trabajadas, las horas de descanso y las horas extraordinarias, realizando un cálculo del salario total.

Añade la opción de enviar por correo electrónico las tarjetas de tiempo para gestionar las horas de trabajo desde el ordenador o enviárselas a quien quieras.

Permite visualizar el registro de un día en particular y editarlo.

<sup>10</sup> iTunes, TimeSheet. Capturas de pantalla del iPhone. [Recuperadas de: <https://itunes.apple.com/es/app/tarjeta-tiempo-timesheet-gestiona/id445811752?mt=8>] [consulta: 3 de junio de 2016]

## 2.2.2. Gestión de clientes



### 2.2.2.1 SugarCRM Mobile [9]

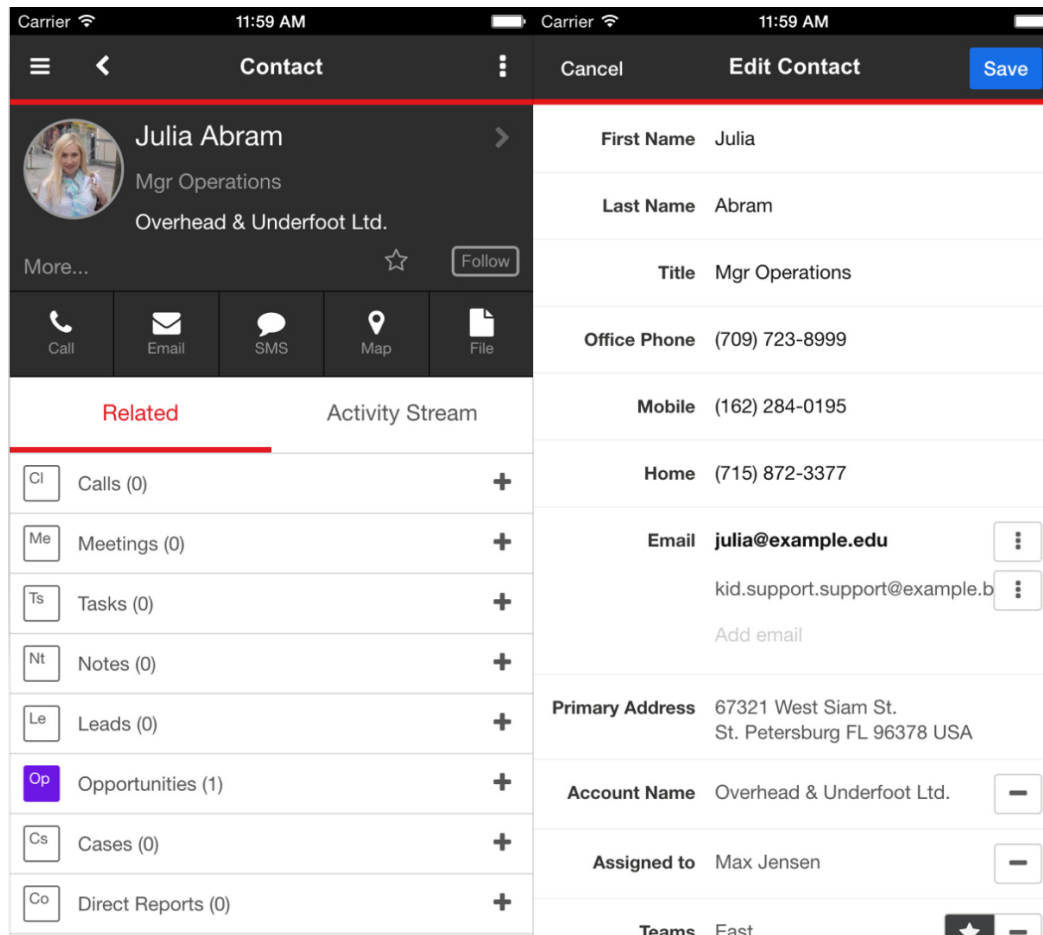


Figura 2.2.2.1.1 Capturas de pantalla del iPhone de la aplicación SugarCRM<sup>11</sup>.

SugarCRM es una aplicación de administración de relaciones con clientes. Permite clasificar los proyectos en *leads*, *opportunities*, etc. y te ofrece la posibilidad de consultar el perfil del contacto de la empresa cliente, con el objetivo de ir preparado a una reunión.

Dentro del perfil del contacto puedes iniciar de manera rápida una llamada, mandar un correo electrónico o enviar un mensaje de texto.

<sup>11</sup> iTunes, SugarCRM. Capturas de pantalla del iPhone. [Recuperadas de: <https://itunes.apple.com/es/app/sugarcrm-mobile/id599208633?mt=8>] [consulta: 3 de junio de 2016]

## 2.2.3 Comunicación



### 2.2.3.1 Noysi [10]

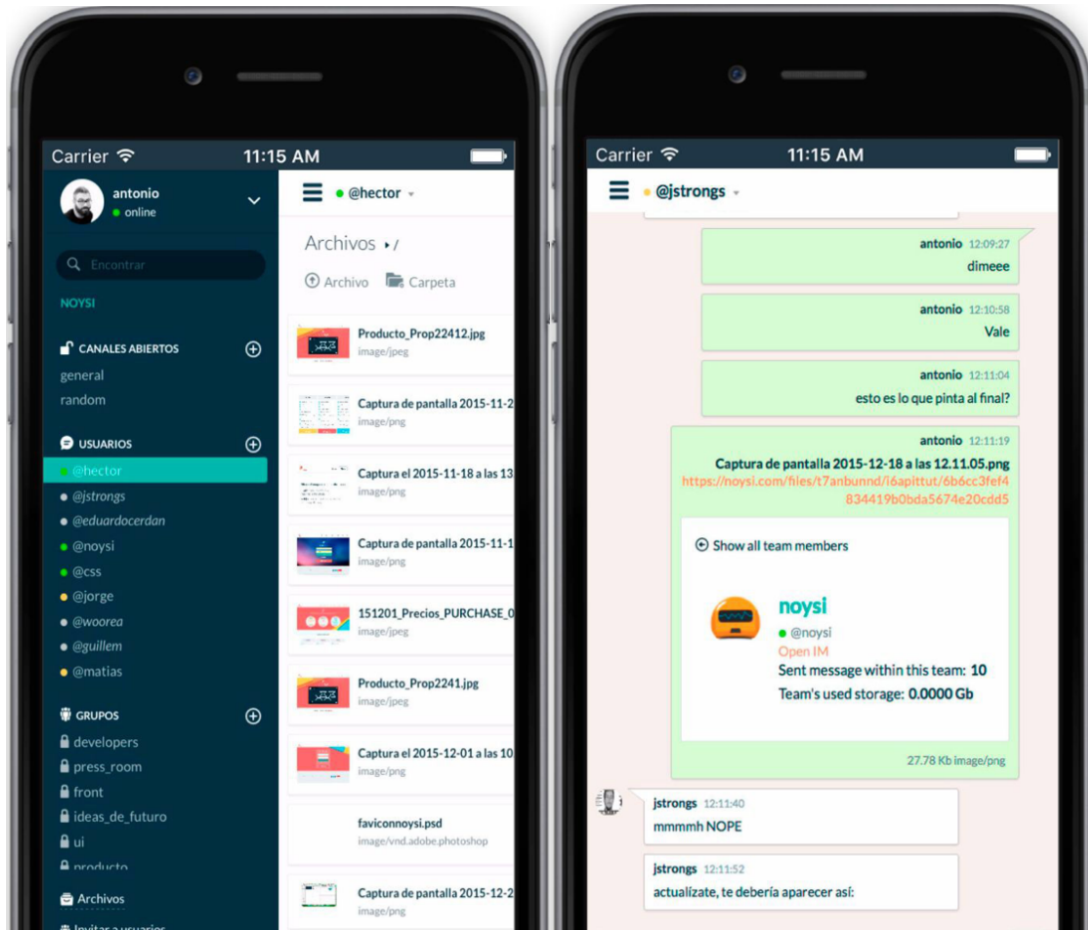


Figura 2.2.3.1.1 Capturas de pantalla del iPhone de la aplicación Noysi<sup>12</sup>.

Noysi es una aplicación cuya funcionalidad principal es la de poner en contacto a los miembros de un equipo.

Una de sus principales ventajas es que es multiplataforma, es decir, tiene también aplicación de escritorio, lo que permite usar el ordenador para iniciar una conversación.

Ofrece un nivel de seguridad similar al que utiliza la banca online y está integrado con las aplicaciones más populares.

<sup>12</sup> iTunes, Noysi. Capturas de pantalla del iPhone. [Recuperadas de: <https://itunes.apple.com/es/app/noysi/id1071527690?mt=8>] [consulta: 3 de junio de 2016]



### 2.2.3.2 Mmmelon [11]

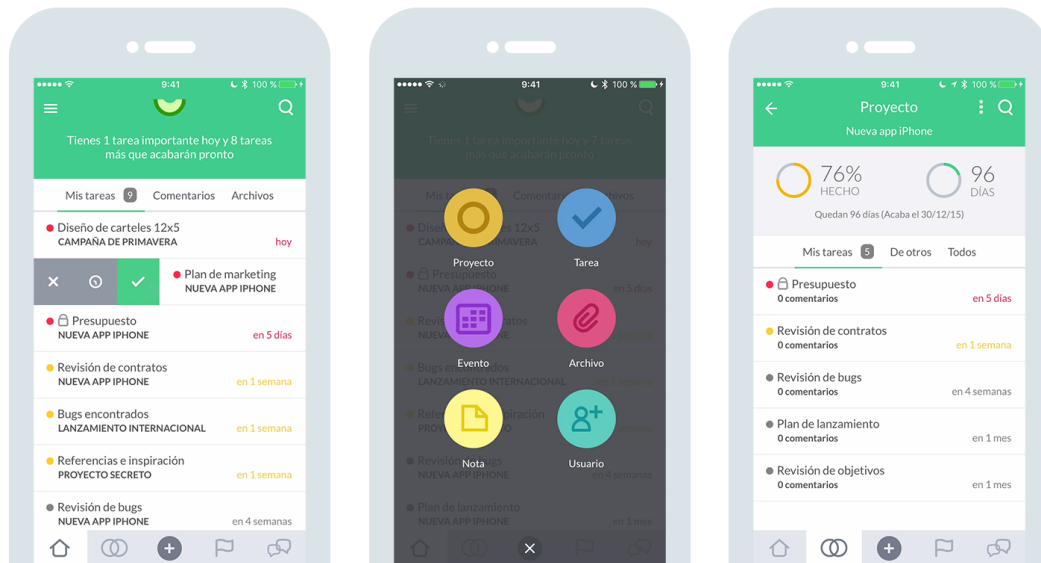


Figura 2.2.3.2.1 Capturas de pantalla del iPhone de la aplicación Mmmelon<sup>13</sup>.

Mmmelon es una aplicación que te permite crear y controlar proyectos y tareas; crear y completar tareas, subir archivos o añadir comentarios; gestionar qué usuarios tienen accesos a tareas y limitar privilegios a participantes externos; compartir archivos; e iniciar chat con cualquier persona que pertenece a la empresa.

<sup>13</sup> iTunes, Mmmelon. Capturas de pantalla del iPhone. [Recuperada de: <https://itunes.apple.com/es/app/mmmelon-comunicacion-y-trabajo/id1004330221?mt=8>] [consulta: 3 de junio de 2016]

## 2.2.4 Gestión de recursos



### 2.2.4.1 MiniKeePass [12]

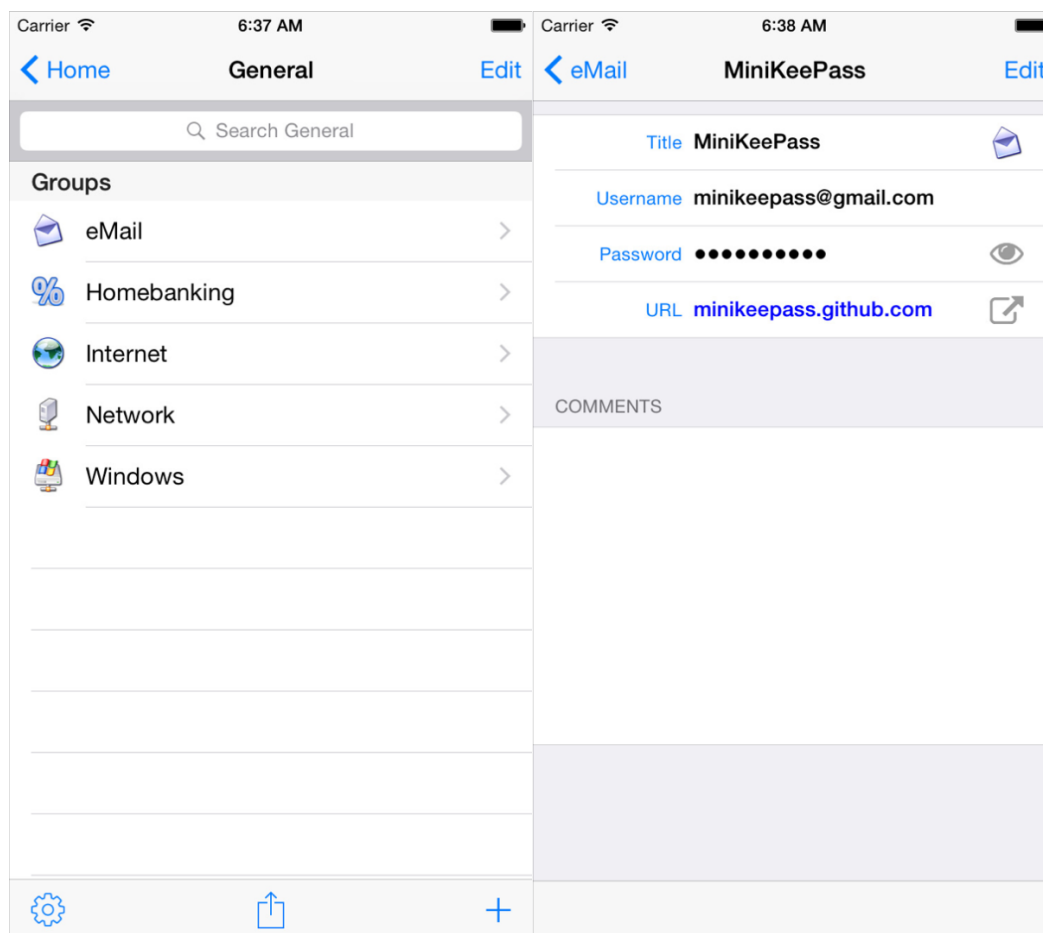


Figura 2.2.4.1.1 Capturas de pantalla del iPhone de la aplicación MiniKeePass<sup>14</sup>.

MiniKeePass es una aplicación que te permite ver, editar y crear archivos que almacenan un título, un usuario, una contraseña, una URL y un comentario cuya finalidad es que no nos olvidemos de dicha contraseña de registro de alguna página web o cualquier otro lugar.

Permite, además, copiar las contraseñas al portapapeles y es una app que puede estar protegida por un código de seguridad e incluso por Touch ID.

<sup>14</sup> iTunes, MiniKeePass. Capturas de pantalla del iPhone. [Recuperadas de: <https://itunes.apple.com/es/app/minikeepass-secure-password/id451661808?mt=8>] [consulta: 3 de junio de 2016]

## 2.3 Conclusiones del análisis

Tras el exhaustivo análisis realizado podemos concluir, por una parte, que la INTRANET es una aplicación que reúne funcionalidades de otras apps encontradas en el mercado, las cuales realizan tareas similares a los módulos de mi aplicación.

Con respecto al control de horas, la mayoría de aplicaciones incluyen una funcionalidad interesante: un cronómetro. Es una de las mejoras que se plantearon al inicio del proyecto y que, tras el estudio de mercado, se terminará desarrollando.

Las aplicaciones encontradas relacionadas con la gestión de clientes son bastante similares al módulo de empresas que podemos encontrar en la INTRANET. De hecho, no cuentan con la posibilidad de acceder a una aplicación de mapas que te guíe hacia la empresa cliente, pero sí que clasifican los proyectos emergentes como *leads* u *opportunities*.

Si hablamos de comunicación, el módulo de mail es bastante más simple que las aplicaciones que existen actualmente, aunque la mayoría de extras no son necesarios en la INTRANET.

Por último, en cuanto a la gestión de recursos, las aplicaciones analizadas incluyen seguridad al tratamiento de los datos de contraseñas, posible mejora que se podría añadir a mi aplicación.

En conclusión, la INTRANET es una app que reúne de forma efectiva las necesidades que pueden tener los empleados y la dirección de una empresa, ayudando a que el día a día sea más llevadero y a tener un seguimiento del rendimiento del trabajador. Pero existen muchas mejoras que pueden hacer de ella una aplicación más completa.



## 2.4 Marco Socio-económico

España es uno de los países de la Unión Europea con mayor grado de aceptación a la telefonía móvil, superando el número de líneas al número de habitantes.

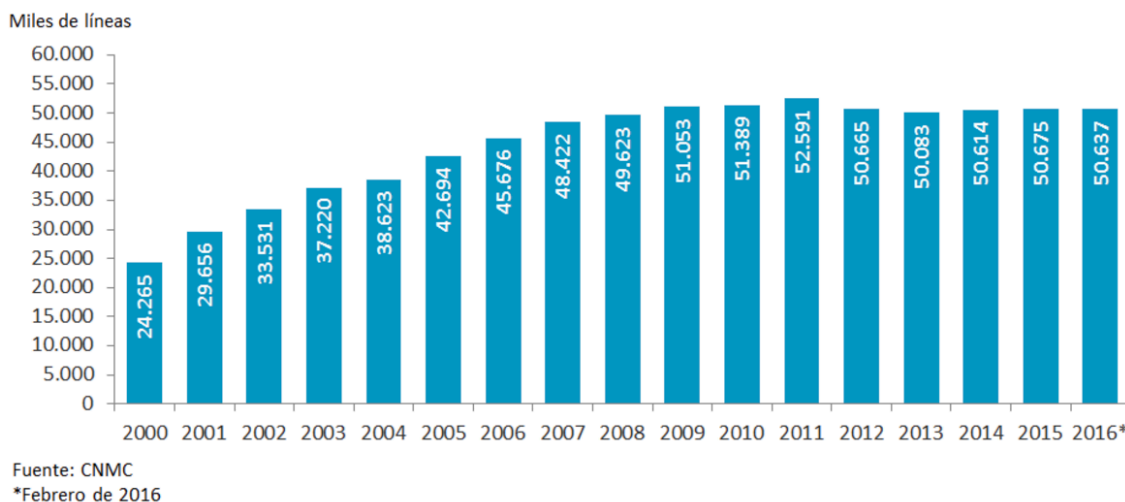


Figura 2.3.1 Número de clientes de telefonía móvil en febrero de 2016<sup>15</sup>.

No es de extrañar entonces que un 96% de la población adulta posea un dispositivo móvil y un 87% un smartphone. Además, un 90% de las personas que poseen un smartphone se conectan a diario [13].

España, una vez más, vuelve a ser líder dentro de la UE en penetración de smartphones. El acceso a Internet se hace cada vez más desde el móvil y la aparición de *wearables* y el *IoT* (*Internet of Things*) demuestra que es un mercado en auge completamente.

La aparición de la banca móvil ha modificado nuestra forma de pago y otros sectores como el marketing, las ventas o la atención al cliente han decidido utilizar estas nuevas tecnologías como medio de difusión [14].

El marco socio-económico augura un buen futuro al mundo de las aplicaciones móviles y anima a seguir investigando nuevas tecnologías que puedan surgir, con el objetivo de no quedarse nunca obsoleto y desarrollar aplicaciones que cubran las nuevas necesidades que aparezcan en el usuario final.

<sup>15</sup> ontsi, "Evolución del número de clientes de telefonía móvil en España", 2016, [en línea]. Recuperada de: <http://www.ontsi.red.es/ontsi/es/indicador/evoluci%C3%B3n-del-n%C3%BAmero-de-clientes-de-telefon%C3%ADa-m%C3%B3vil-en-espa%C3%B1a>

## 2.5 Política de privacidad y tratamiento de datos personales

Con la creciente penetración de las apps en nuestra vida cotidiana se ha hecho necesaria una modificación en la política de privacidad y tratamiento de datos personales puesto que la mayoría de las aplicaciones solicitan nuestra información en numerosas ocasiones.

Los propios desarrolladores son los primeros que desconocen estas leyes y la importancia de tratar bien los datos que los usuarios suministran a la aplicación, en muchas ocasiones de forma desconocida o no deseada.

La información que suelen aportar los usuarios son datos de localización, de contacto, de identificación única del fabricante (IMEI), de identificación del teléfono, de tarjetas de crédito y medios de pago, de historial del navegador o el email, entre otros.

En numerosas ocasiones son datos indispensables para el correcto funcionamiento de la aplicación y para sacarle el máximo partido por lo cual hay que tener especial cuidado. Por ello hay que asegurarle al usuario final una serie de garantías legales para poder procesar sus datos.

Es muy importante contar con su consentimiento, antes de la instalación y, por tanto, del procesado de la información; informar al usuario acerca de qué información concreta se le va a solicitar y qué se va a hacer con ella; que el usuario tenga claro cómo se van a tratar sus datos, sin ambigüedades; y tener claro que el consentimiento libre implica que el usuario final puede aceptar o rechazar el procesamiento de sus datos.

Toda esta información se puede encontrar de manera más detallada en el *Dictamen 2/2013 sobre aplicaciones para móviles inteligentes* que el Grupo Europeo de Protección de Datos del Artículo 29 elaboró el 27 de febrero de 2013 [15].

## 3 Descripción y organización

### 3.1 Descripción de la aplicación a desarrollar

#### 3.1.1 Contexto

El proyecto nace de una aplicación web existente en *Armadillo Amarillo*, llamada INTRANET. La idea era desarrollar una aplicación móvil, de características similares a la ya mencionada web, en las dos plataformas más comunes: iOS y Android.

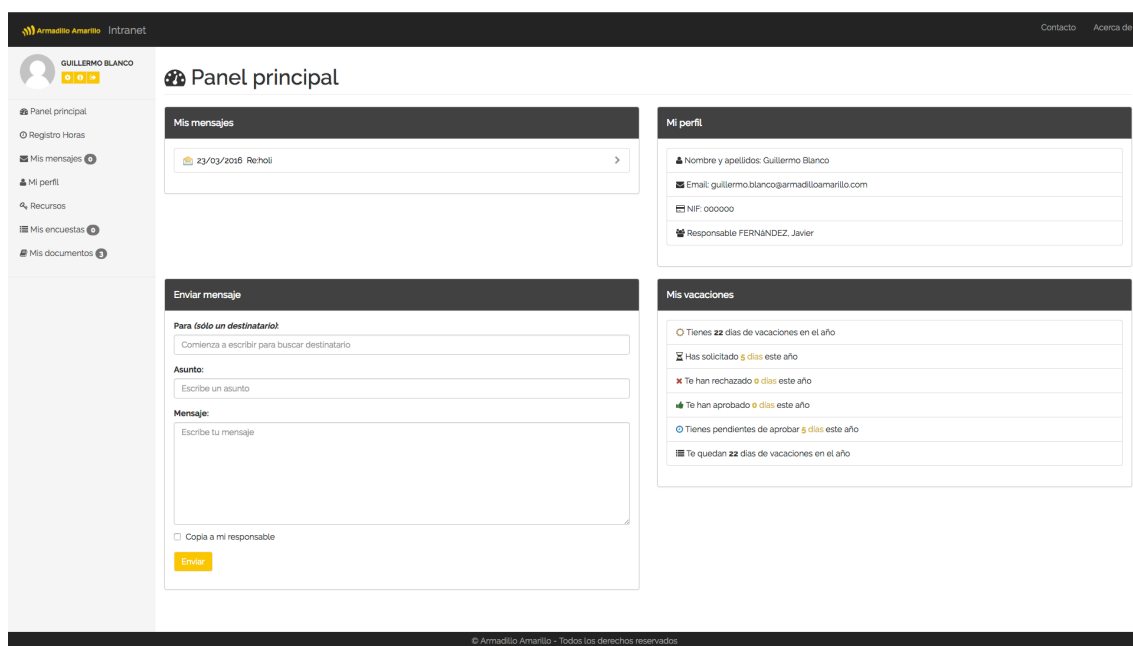


Figura 3.1.1.1 Aplicación web de Armadillo Amarillo.

También existía un prototipo de aplicación móvil que se había empezado a desarrollar con el *framework Ionic*. Este *framework* permite programar, con un solo código, aplicaciones para ambas plataformas.

Era un prototipo sencillo, que constaba de un menú lateral y un calendario para imputar horas, con un cronómetro para calcular el tiempo que le dedicabas a cada proyecto. El resto de módulos estaba sin desarrollar.

La aplicación web estaba ligada a un *backend*, programado en php, que tenía una estructura con los módulos y los servicios necesarios para que la web funcionase.

▼ modules	▶ history	▶ sales_contact
▶ calendar	▶ job	▶ sales_lead
▶ calendar_item_type	▶ lang	▶ sales_lead_state
▶ candidate	▶ learning	▶ sales_lead_type
▶ company	▶ login	▶ solutions
▶ contact	▶ mail	▶ statistics
▶ course	▶ main	▶ tag
▶ course_catalogue	▶ news	▶ tag_activity
▶ course_catalogue_category	▶ notifications	▶ tag_teaching_unit
▶ course_catalogue_type	▶ permissions	▶ teaching_unit
▶ course_doc	▶ products	▶ teaching_unit_dependence
▶ course_teaching_unit	▶ question	▶ time_track
▶ doc	▶ quiz	▶ time_track_type
▶ error	▶ quiz_question	▶ tutor_course
▶ exam	▶ recruitment	▶ user_course
▶ exam_answer	▶ recruitment_cv	▶ user_documentation
▶ exam_question	▶ recruitment_job	▶ user_exam
▶ exam_question_type	▶ recruitment_state	▶ user_exam_question
▶ expenses	▶ resource	▶ user_file
▶ expenses_type	▶ resource_group	▶ user_quiz
▶ experience	▶ resource_type	▶ user_quiz_question
▶ file	▶ role	▶ users
▶ file_type	▶ role_permissions	▶ users_resource

Figura 3.1.1.2 Módulos del *backend*.

El *backend* incluía una base de datos *sql* con las tablas hechas.

TABLAS	n_doc	n_resource_type
k_permissions	n_exam	n_sales_contact
k_role	n_exam_answer	n_sales_lead
k_role_permissions	n_exam_question	n_sales_lead_state
k_services	n_exam_question_type	n_sales_lead_type
k_sessions	n_expenses	n_tag
k_users	n_expenses_type	n_tag_activity
n_activity	n_experience	n_tag_question
n_calendar	n_file	n_tag_teaching_unit
n_calendar_item_type	n_file_type	n_tag_test
n_candidate	n_history	n_teaching_unit
n_company	n_job	n_teaching_unit_dependence
n_company_doc	n_log	n_test
n_course	n_mail	n_time_track
n_course_catalogue	n_news	n_time_track_type
n_course_catalogue_category	n_notifications	n_tutor_course
n_course_catalogue_type	n_post	n_user_course
n_course_doc	n_question	n_user_documentation
n_course_mail	n_quiz	n_user_exam
n_course_teaching_unit	n_quiz_question	n_user_exam_question
n_doc	n_recruitment	n_user_file
n_exam	n_recruitment_cv	n_user_quiz
n_exam_answer	n_recruitment_job	n_user_quiz_question
n_exam_question	n_recruitment_state	n_users_resource
n_exam_question_type	n_resource	t_customers
	n_resource_group	t_pages

Figura 3.1.1.3 Tablas de la base de datos.

Partiendo de esta base, yo me encargaría del desarrollo de la aplicación **iOS** en nativo, es decir, en el lenguaje propio de *Apple*, *Objective-C*, y un compañero se encargaría del desarrollo de la aplicación **Android** en nativo también.

A pesar de que ambos compartiríamos la misma idea de aplicación y, a grandes rasgos, un diseño parecido, el desarrollo sería completamente independiente ya que ambos lenguajes son diferentes y no tienen nada que ver.

Por otro lado, ambos atacaríamos contra el mismo *backend* y utilizaríamos los mismos servicios, llamando a las mismas funciones, por lo cual realizamos un reparto de módulos y cada uno se ocuparía del desarrollo de servicios diferentes en el servidor.

### 3.1.2 Especificación de requisitos

La aplicación debe ser parecida a la página web actual de la INTRANET, que consta de 5 módulos: Panel Principal, Registro de Horas, Correo, Mi Perfil y Recursos. Además la idea era añadir uno nuevo llamado Compañías.

La funcionalidad de los módulos también era similar y se buscaba que ambas aplicaciones, la web y la móvil, interactuasen entre sí.

El Panel principal es un resumen de los tres últimos correos, registros de horas y recursos del usuario.

El módulo de Registro de Horas ayuda a los empleados a controlar el tiempo que invierten en cada proyecto.

Correo es un módulo que permite a los trabajadores comunicarse entre sí.

El módulo de Mi Perfil contiene información personal de usuario y permite realizar ciertos cambios.

Uno de los módulos más interesantes es el de Recursos, el cual sirve para compartir de manera sencilla y ágil usuarios y contraseñas que puedan interesar a ciertos empleados.

El último módulo es el de Compañías, que alberga información de las empresas con las que se está trabajando.

### 3.1.3 Descripción de la aplicación móvil y wearable

**INTRANET** es una aplicación de gestión para PYMEs (Pequeñas Y Medianas Empresas). Está pensada para facilitar el trabajo tanto a los empleados de una empresa como a la dirección. Consta de varios módulos que se describen a continuación:

Cuando iniciamos la aplicación, aparece una ventana de **Login**, en la cual debemos introducir nuestras credenciales, en este caso, el email de la empresa y la contraseña. Si pulsamos el botón Login, tras rellenar dichos campos, la aplicación muestra el panel principal. Saltan alertas en el caso de que haya algún campo sin rellenar, en el caso de que no exista el usuario o que el usuario y la contraseña no coincidan y en el caso en el que la conexión falle.

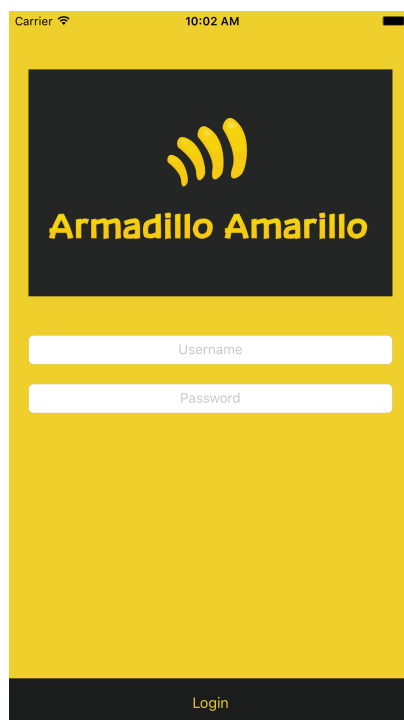


Figura 3.1.3.2 Pantalla de la aplicación móvil del módulo de Login.

Para navegar entre los diferentes módulos utilizaremos un **Menú Lateral**, en el cual aparecen todos ellos. En orden: Panel Principal, Registro de Horas, Mis Mensajes, Mi Perfil, Recursos y Empresas. También muestra tu foto de perfil y tu nombre.

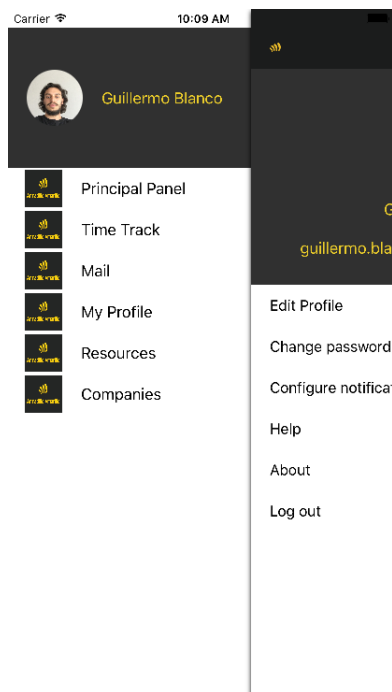
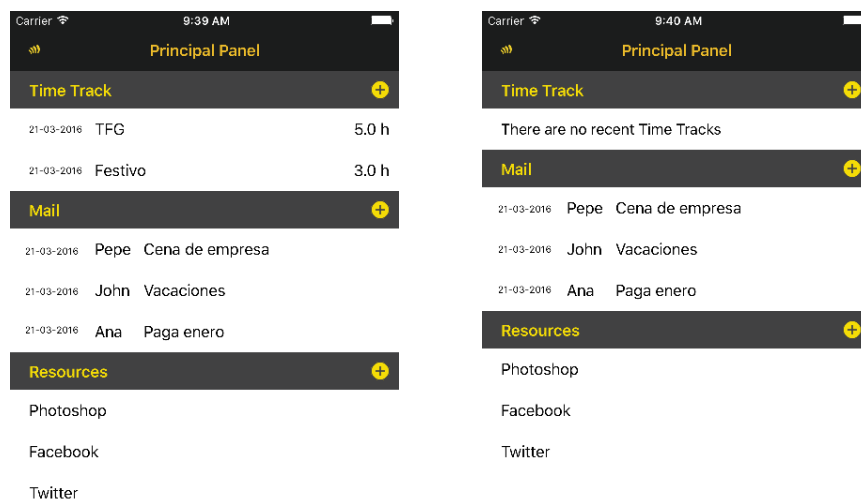


Figura 3.1.3.2 Pantalla de la aplicación móvil del módulo Menú Lateral.

El **Panel Principal** muestra un resumen con los tres últimos registros de horas que se han imputado, los tres últimos mensajes que se han recibido y los tres últimos recursos que se han añadido. Desde él se puede imputar un nuevo registro, enviar un nuevo mensaje y añadir un nuevo recurso. En el caso de que no haya ningún mail, recurso o imputación de horas mostrará un mensaje informando de que no existen.



**Figura 3.1.3.3 Pantallas de la aplicación móvil del módulo Panel Principal.**



El módulo de **Registro de Horas** es uno de los más complejos. Muestra un calendario para poder acceder a un listado de los registros que se han imputado cualquier día. También permite añadir nuevas imputaciones, consultar las que ya existen, modificarlas y borrarlas.

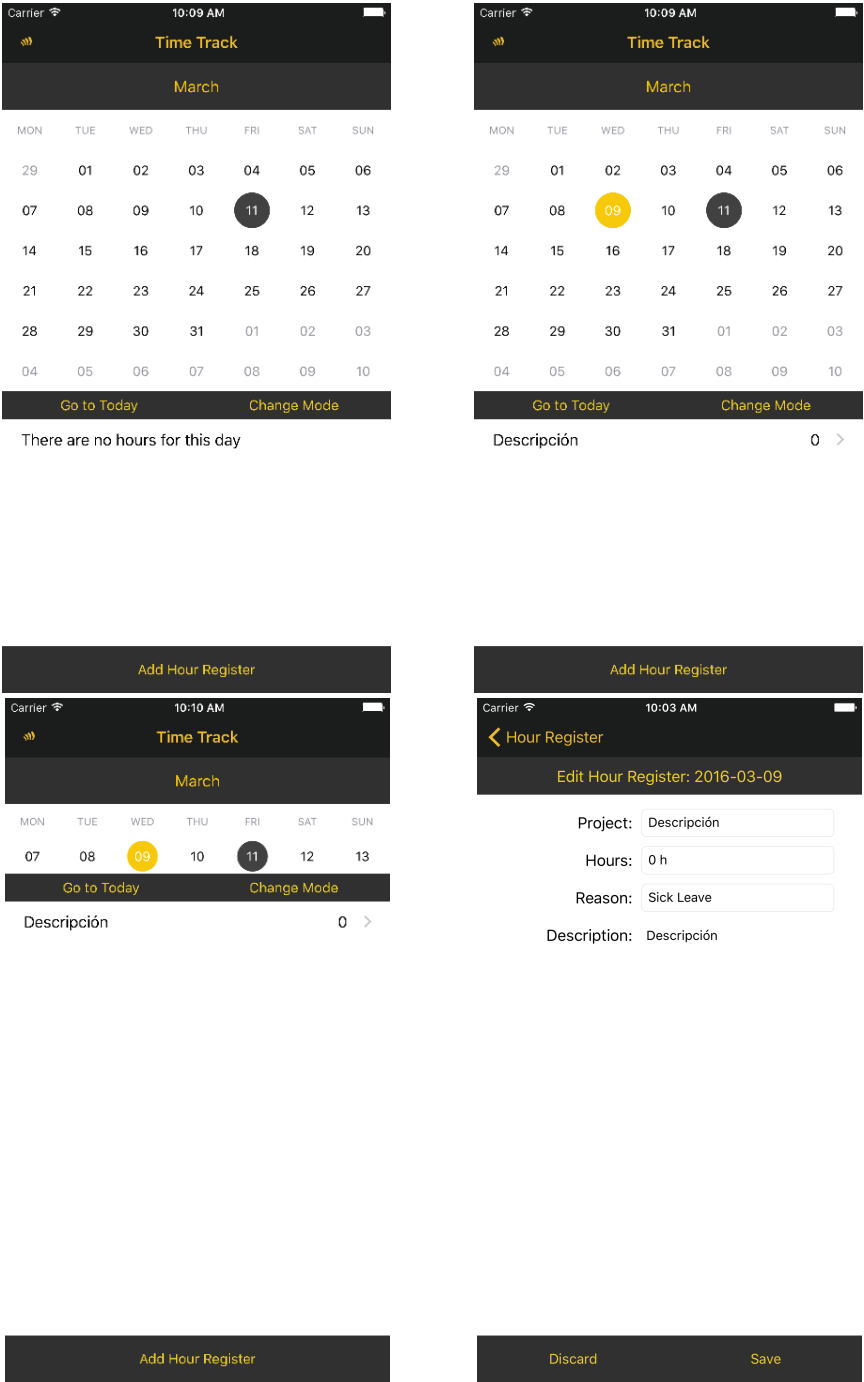


Figura 3.1.3.4 Pantallas de la aplicación móvil del módulo Registro de Horas.

El tercer módulo es el de **Mis Mensajes**. Lo primero que vemos cuando lo seleccionamos es un listado con los mensajes de la bandeja de entrada. Un botón en la parte inferior nos permite mostrar los de la bandeja de salida. Podemos ver con detalle un mensaje, responder y escribir uno nuevo.

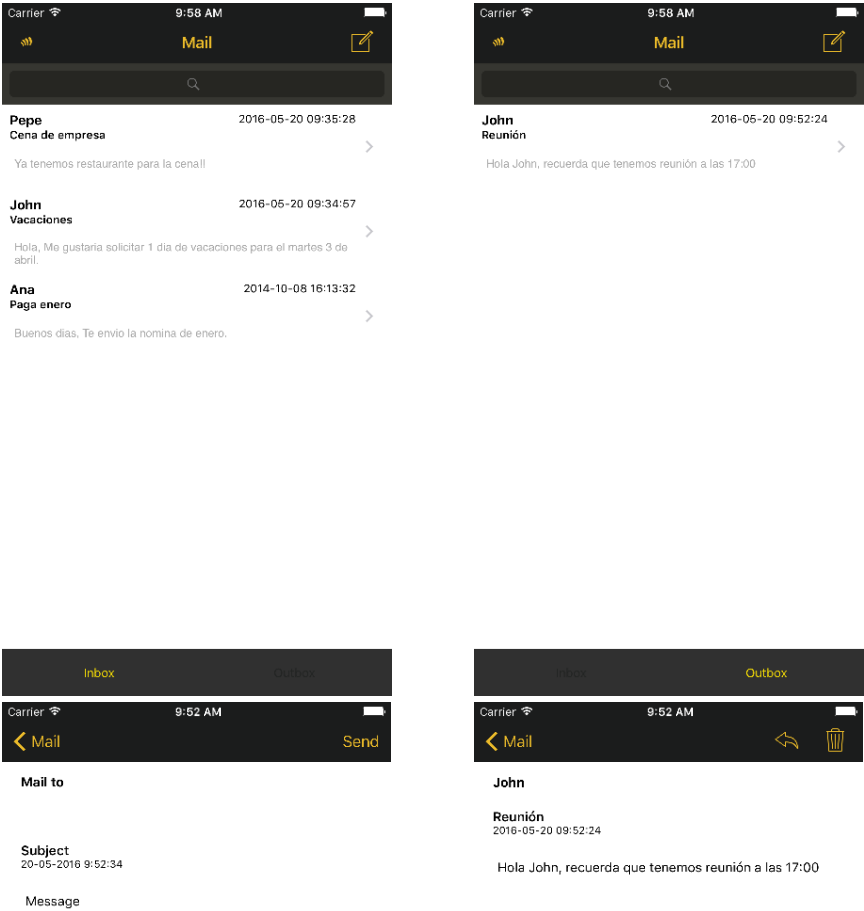
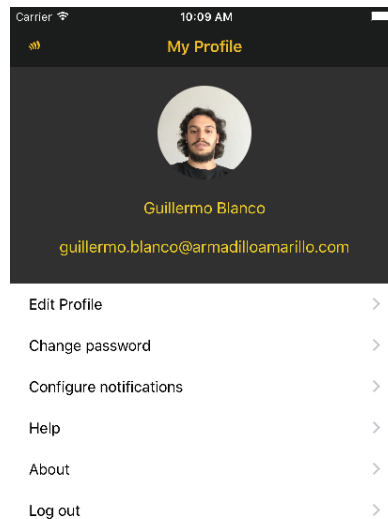


Figura 3.1.3.5 Pantallas de la aplicación móvil del módulo Correo.

**Mi Perfil** es un módulo muy común en la mayoría de las aplicaciones. En nuestro caso es una vista que muestra una foto de perfil, el nombre y apellidos y el email de la empresa. Justo debajo de esa información aparece una lista, en la cual podemos seleccionar varias acciones.



**Figura 3.1.3.6** Pantalla de la aplicación móvil del módulo Mi Perfil.

La primera es editar perfil, que nos llevará a una vista en la que podemos modificar algunos de los datos personales como el nombre, apellido, email o el NIF; la segunda pestaña nos permite cambiar la contraseña; la siguiente es la pestaña de configurar notificaciones, en la cual podemos activar o desactivar las notificaciones *push* y las notificaciones por correo electrónico; las dos siguientes pestañas son las de “ayuda” y “acerca de”; la última es la pestaña de cerrar sesión, que nos devuelve a la ventana de Login del comienzo.



Figura 3.1.3.7 Pantallas de la aplicación móvil del módulo Ajustes de Mi Perfil.

El siguiente módulo que aparece en el Menú Lateral es **Recursos**. Cuando lo pulsamos nos aparece una lista de los grupos de recursos a los que tenemos acceso. Cuando seleccionamos uno, aparece una lista con todos los recursos de ese grupo y ofrece la posibilidad de consultarlos, modificarlos, borrarlos y añadir uno nuevo.

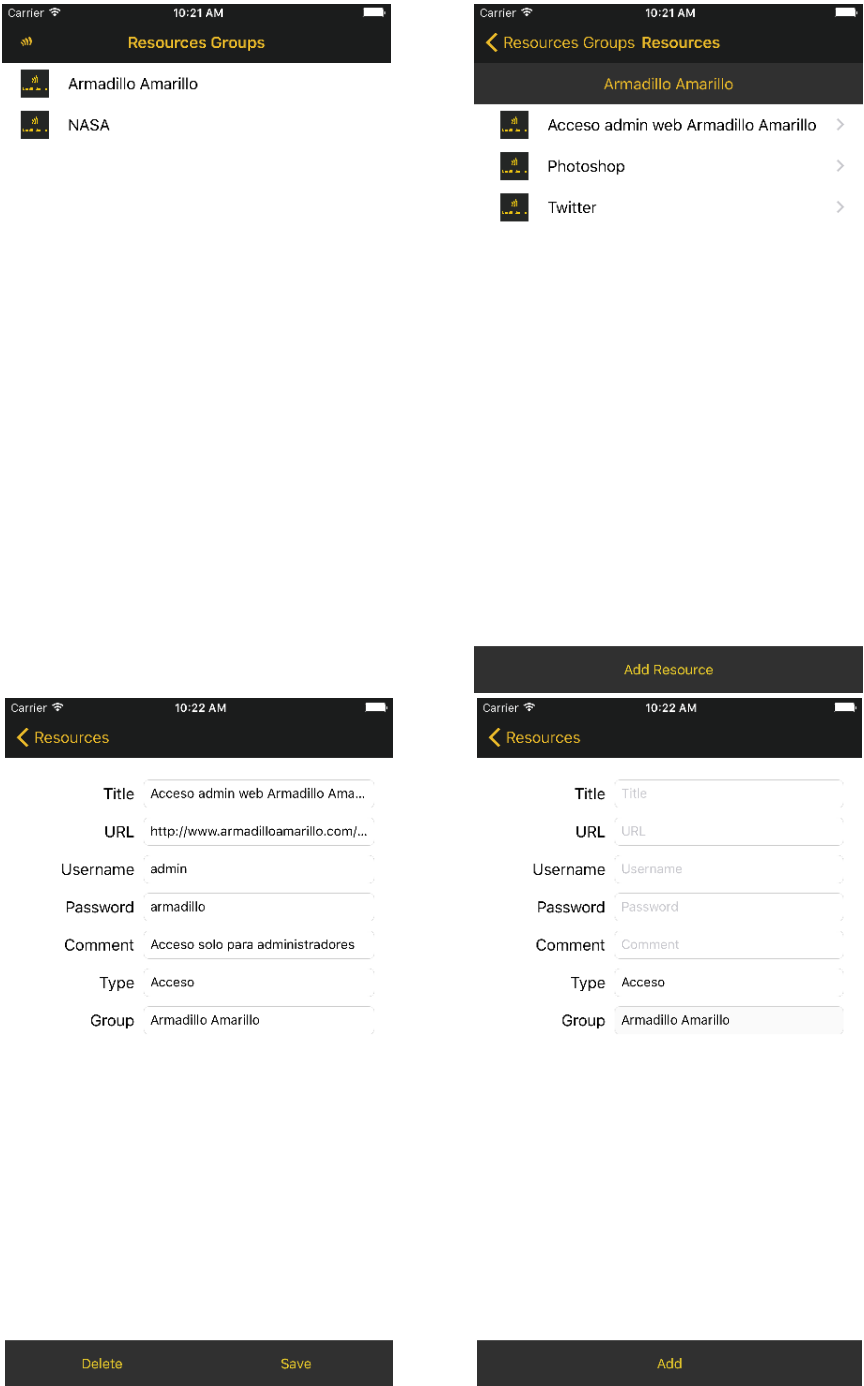


Figura 3.1.3.8 Pantallas de la aplicación móvil del módulo Recursos.

El último de los módulos a los que podemos acceder desde el Menú Lateral es el de **Empresas**. Nos ofrece un listado con las empresas cliente y si pulsamos sobre una de ella nos muestra una lista con las personas contacto de esa empresa. Entonces podremos acceder a sus datos personales como el nombre, su número de teléfono, su correo o el puesto que desempeña en la empresa. Si pulsamos sobre el número de teléfono o el correo se abre la aplicación del teléfono para realizar una llamada o la aplicación del mail para enviar un correo electrónico respectivamente.

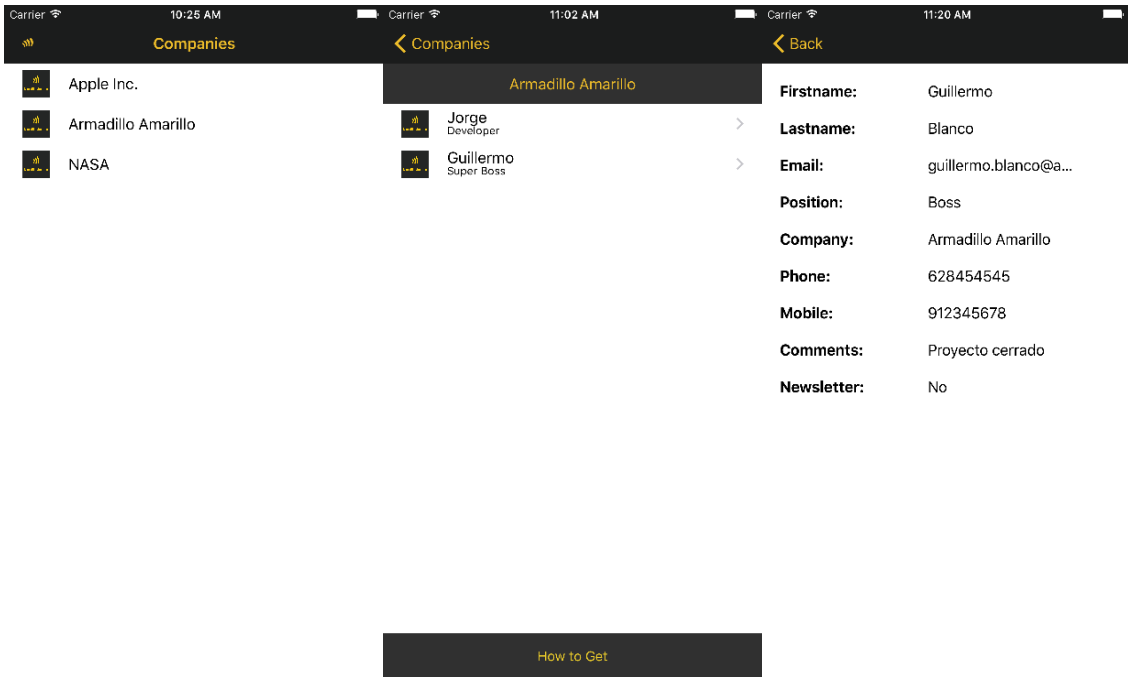


Figura 3.1.3.9 Pantallas de la aplicación móvil del módulo Empresas.

Por último, en la parte inferior de la vista de los contactos de una empresa existe un botón de “Cómo llegar”. Si lo pulsamos nos ofrece la posibilidad de elegir una de las aplicaciones de mapas que tengamos instaladas para que calculen una ruta y saber llegar a la empresa.

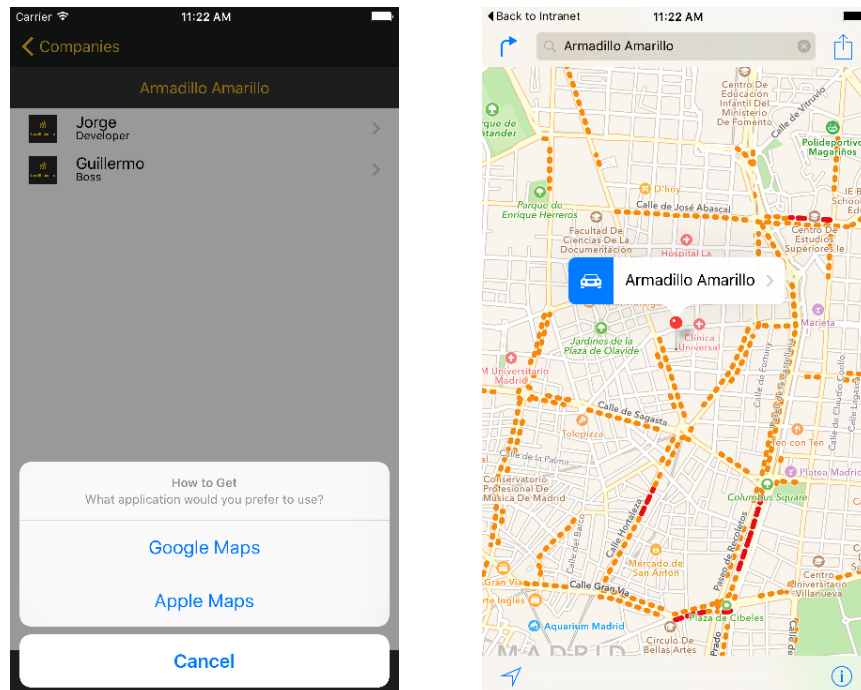


Figura 3.1.3.10 Pantallas de la aplicación móvil del módulo Cómo Llegar de Empresas.

La aplicación **wearable** mostrará una notificación cuando llegue un nuevo mensaje. La alerta se verá como se muestra a continuación:

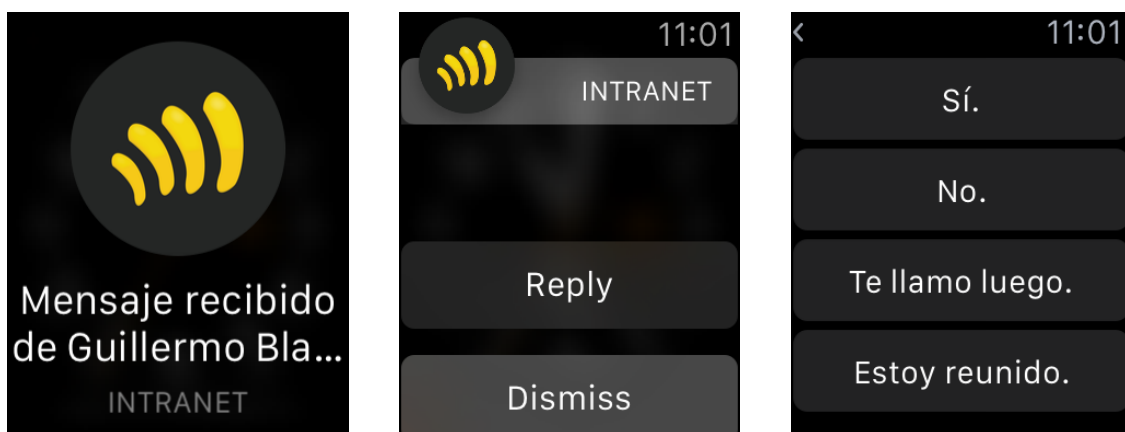


Figura 3.1.3.11 Pantallas de la aplicación **wearable**.

Desde el dispositivo Apple Watch puedes responder al mensaje que te ha llegado con unas frases predefinidas.

### 3.1.4 Características del usuario objetivo

El usuario final de esta aplicación será una persona que trabaje en una pequeña o mediana empresa y que posea un iPhone con una versión igual o superior a iOS 8.0.

Por ello, el margen de edad de personas que usarán esta aplicación está comprendido entre la edad mínima para trabajar, es decir, a los 16 años en España, según la normativa vigente y, como no existe un máximo de edad, hasta que dicha persona decida jubilarse.

### 3.1.5 Restricciones de diseño

El uso de *constraints* facilita sobremanera el problema de que existan iPhones con diferentes tamaños, ya que te permite estandarizar el diseño de las vistas para que, por ejemplo, un botón esté siempre centrado o a una cierta distancia de otro elemento de diseño que tenga al lado.

La única restricción de diseño que ha aparecido es que las pantallas de los modelos de iPhone 5, 5s y SE son considerablemente más estrechas que las de sus hermanos mayores y, por ello, se ha tenido que adaptar el tamaño de ciertos elementos de diseño a estas pantallas de menor dimensión.

Por otro lado, para poder instalar la app será necesario disponer del espacio suficiente en la memoria del dispositivo.

La aplicación se ha optimizado, no obstante, el rendimiento dependerá de la potencia del procesador y demás características del dispositivo en el que se instale.



### 3.2 Estimación del proyecto y presupuesto

La estimación del proyecto se realizó al comienzo del proyecto. Se tuvieron en cuenta las horas que se iban a dedicar al diseño de las vistas, a la programación y funcionalidades, a las modificaciones y ampliaciones del backend y a la modificación de las tablas de la base de datos.

Como en cualquier estimación, los cálculos son orientativos y es normal desviarse de lo previsto.

Pantalla	Horas Diseño app	Horas Programación app	Horas Backend	Horas DDBB
Login	4	6		
Menú lateral	8	8		
Panel Principal	4	12	2	
Registro Horas (Pop-up Eliminar)	8	16	1	
Editar Día	2	4	1	
Mensajes	8	16	4	
Redactar/vista Mensaje	6	12		
Recursos	6	12	2	
Ver/Modificar/Crear Recurso	4	8	1	
Cómo Llegar	6	14	2	1
Facturas	4	4	8	
Crear/Modificar Factura	2	4	2	
Mi Perfil	6	6		
Editar Perfil	6	20	6	
Notificaciones	2	12	16	1
Acerca De y Ayuda	2	0		
Wearable	12	12	4	
Subtotal horas	90	166	49	
Subtotal Días	18	33,2	9,8	
Total Horas	305			
Total Días	61			
Total Meses	2,77			

Tabla 3.2.1 Estimaciones en horas del proyecto.

A continuación se hará un cálculo del presupuesto total del proyecto, incluyendo horas empleadas en el diseño, el desarrollo tanto de la aplicación móvil como la aplicación *wearable*, el desarrollo del *backend* y la redacción de la memoria.

También se tendrá en cuenta todo el trabajo anterior al inicio del proyecto porque, como ya ha sido comentado en puntos anteriores, el *backend* estaba ya creado así como la base de datos.

Horas	Tarea	Cargo	Precio/hora	Subtotal
40	<i>Backend</i>	Ingeniero sénior	50,00 €	2.000,00 €
10	Diseño aplicación	Ingeniero junior	35,00 €	350,00 €
305	Desarrollo aplicación	Ingeniero junior	35,00 €	10.675,00 €
75	Redacción memoria	Ingeniero junior	35,00 €	2.625 €
			<b>TOTAL</b>	<b>15.650,00 €</b>

Tabla 3.2.2 Costes de Personal

Concepto	Precio	Amortización	Importe
MacBook	1.279,00 €	¼	319,75 €
RAM + SSD	250,00 €	¼	62,5 €
iPhone 5 16Gb	549,00 €	¼	137,25 €
iPhone 6s 16Gb	749,00 €	¼	187,25 €
Apple Watch 38mm	669,00 €	¼	167,25 €
Microsoft Office 365 Empresa Essentials	4,20 €	-	4,20 €
Xcode 7.3	0,00 €	-	0,00 €
		TOTAL	878,20 €

Tabla 3.3.3 Costes de material

Concepto	Importe
<b>Costes de personal</b>	15.650,00 €
<b>Costes de material</b>	878,20 €
<b>Costes indirectos (20%)</b>	3.305,64 €
<b>Base imponible</b>	19.833,84 €
<b>I.V.A. (21%)</b>	4.165,11 €
<b>TOTAL</b>	<b>23.998,95 €</b>

Tabla 3.3.3 Presupuesto total

### 3.3 Metodología de trabajo

#### 3.3.1 SCRUM

SCRUM es una metodología de trabajo que se basa en el seguimiento de unas buenas prácticas para que un proyecto salga adelante. Estas buenas prácticas están enfocadas al trabajo en equipo.

Se realizan una serie de entregas parciales del proyecto final, con el objetivo de identificar situaciones en las que el cliente no está contento, existe un cierto retraso, los costes han superado lo estipulado o la calidad no es buena.

Este tipo de metodología encaja perfectamente en entornos cuyos proyectos son complejos, es decir, proyectos que tienen que estar listos en poco tiempo, cuyos requisitos son cambiantes y donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

En mi caso, se han utilizado tres herramientas para seguir esta metodología: el reparto de tareas en sprints, SourceTree y Jira.

Los **sprints** son una especie de hitos que tienen una duración estimada, en nuestro caso 2 semanas, y un día de entrega determinado. Estos sprints se subdividen en tareas o historias de usuario, que son, como su propio nombre indica, tareas que hay que hacer para que el desarrollo de la aplicación salga adelante. Varias tareas se pueden unir y formar parte de una historia épica, que nada tiene que ver con un sprint, puesto que un sprint puede englobar varias historias épicas o coger parte de una.

Para la asignación de tareas se utiliza el gestor de proyectos **Jira**, en el cual, el *Project Manager* elige qué tareas le asigna a cada programador. Una vez tenemos las tareas asignadas, podemos elegir entre cinco columnas para colocarlas: *To Do*, para las tareas que te han asignado; *In Progress*, para las tareas asignadas que están en proceso de desarrollo; *Blocked*, para las tareas bloqueadas, *Waiting for QA*, para las tareas finalizadas que están a la espera de estar evaluadas; y *Done*, para las tareas finalizadas y comprobadas.

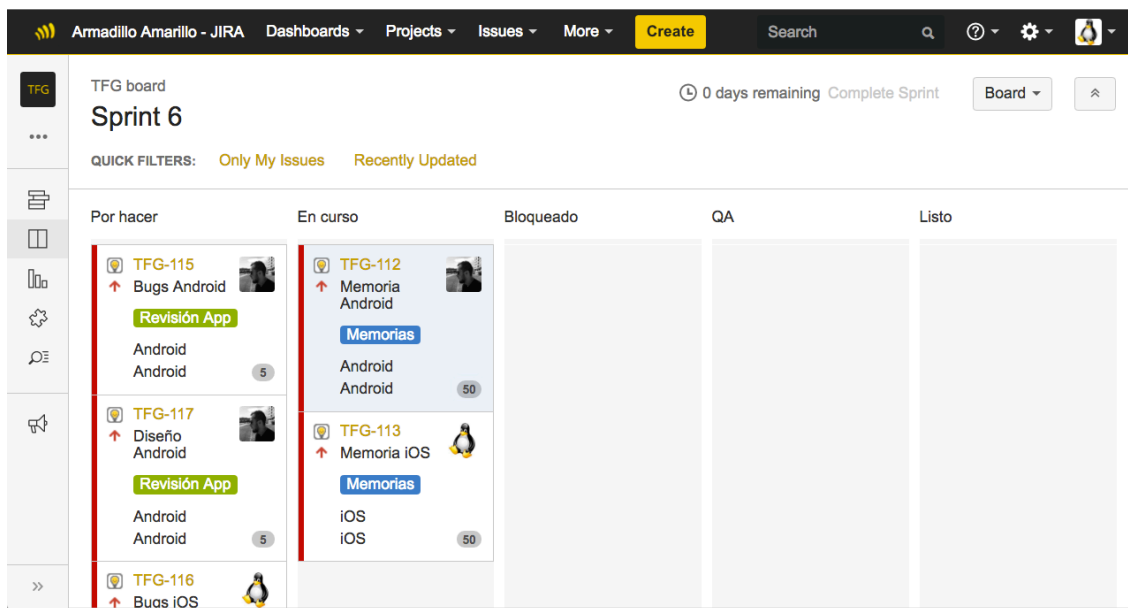


Figura 3.3.1.1 Ejemplo de Historias de Usuario en Jira.

De esta forma es muy fácil llevar un control de las tareas que se van completando y se consigue que dos personas no estén ocupando su tiempo con el mismo trabajo. Además se obtiene una visión más fiable del avance del proyecto.

Por último, los programadores hacemos uso de un software de control de versiones (*Git*), llamado **SourceTree**<sup>16</sup>, que permite ir guardando y actualizando los cambios que se hacen en un entorno local para evitar perder el trabajo realizado y para tener la posibilidad de recuperar una versión anterior del desarrollo. Todas estas actualizaciones se ponen en común en una rama maestra para que todo el mundo pueda acceder a ella y continuar el proyecto donde lo dejó su compañero anterior.

Graph	Description	Commit	Author	Date
	Uncommitted changes	*	*	hoy 19:39
	origin/develop Merge branch 'develop' of gitlab.armadilloamarillo.com:armadilloamarillo/tfg_ios into develop	4338af1	Guillermo Blanco...	hoy 4:43
	HEAD Deleted unused directories.	ae8bb2b	Guillermo Blanco...	31 may 2016 9:35
	merge conflicts	aa82780	Guillermo Blanco...	hoy 4:41
	Some changes.	f91381d	Guillermo Blanco...	24 may 2016 9:04
	Contact sale view modified	6a105fb	Guillermo Blanco...	20 may 2016 12:27
	Merge branch 'develop' of gitlab.armadilloamarillo.com:armadilloamarillo/tfg_ios into develop	bf911d9	Guillermo Blanco...	19 may 2016 13:34
	show message from notification	66f603d	Guillermo Blanco...	18 may 2016 22:23
	no message	bf9222a	Guillermo Blanco...	19 may 2016 13:23
	Some changes on Watch App.	0fd067d	Guillermo Blanco...	10 may 2016 14:40
	Working Intranet in new iMac	f3cd0f2	Guillermo Blanco...	5 may 2016 11:05
	Watch integrated. Push working correctly.	e50ae4d	Guillermo Blanco...	5 may 2016 9:22
	Push Notifications work.	68d71a2	Guillermo Blanco...	3 may 2016 11:21
	origin/feature/TFG_IOS_Watch Post request problems solved.	0a0d2e1	Guillermo Blanco...	20 abr 2016 12:30
	Post Request Done.	adca9ca	Guillermo Blanco...	19 abr 2016 10:33
	Now with changes.	0814e75	Guillermo Blanco...	15 abr 2016 13:17
	Watch integrated. Init Push Notifications.	3181bf7	Guillermo Blanco...	15 abr 2016 13:16
	First Commit wearable.	e1f022f	Guillermo Blanco...	11 abr 2016 12:44
	Removed file that couldnt be tracked	07a6ab5	Guillermo Blanco...	11 abr 2016 11:21
	origin/feature/TFG_IOS_Camera TimeTrack problem resolved. Just upload photo into ddbb after change post mode.	74a6b54	Guillermo Blanco...	11 abr 2016 11:15
	Camera works ok. Just select Photo from library and save photo in ddbb.	bc38b4a	Guillermo Blanco...	7 abr 2016 11:48
	Principal Panel Done. Start with camera.	722693b	Guillermo Blanco...	6 abr 2016 13:19
	origin/feature/TFG_IOS_Mail Mail View done after laptop crushed on update.	02c93b5	Guillermo Blanco...	5 abr 2016 10:14
	Company Done. Mail does not work.	fe47971	Guillermo Blanco...	30 mar 2016 13:25

Figura 3.3.1.2 Ejemplo de un proyecto en SourceTree.

<sup>16</sup> SourceTree. Disponible en: <https://www.sourcetreeapp.com/>

### 3.3.2 Historias de usuario

#### 3.3.2.1 Sprint 1

Nombre	Diseño Login	ID	TFG-7
Sprint	1	Épica	Login/Registro
Fecha creación	05/Feb/2016	Fecha realización	10/Mar/2016
Descripción	<ul style="list-style-type: none"><li>Diseño del logotipo de la empresa, campo de usuario, campo de contraseña y botón Login.</li></ul>		

Tabla 3.3.2.1.1 Historia de usuario Diseño Login.

Nombre	Hacer Login	ID	TFG-8
Sprint	1	Épica	Login/Registro
Fecha creación	05/Feb/2016	Fecha realización	10/Mar/2016
Descripción	<ul style="list-style-type: none"><li>Conexión introduciendo el usuario y la contraseña correctos.</li></ul>		

Tabla 3.3.2.1.2 Historia de usuario Hacer Login.

Nombre	Gestión de errores Login	ID	TFG-9
Sprint	1	Épica	Login/Registro
Fecha creación	05/Feb/2016	Fecha realización	10/Mar/2016
Descripción	<ul style="list-style-type: none"><li>Probar qué ocurre si dejamos algún campo vacío, los dos vacíos, la contraseña incorrecta o usuario inexistente.</li><li>Fallo de conexión para toda la aplicación.</li></ul>		

Tabla 3.3.2.1.3 Historia de usuario Gestión de errores Login.

Nombre	Diseño Menú Lateral	ID	TFG-13
Sprint	1	Épica	Menú Lateral
Fecha creación	05/Feb/2016	Fecha realización	10/Mar/2016
Descripción	<ul style="list-style-type: none"><li>Generar vista de diseño de Menú Lateral, que mostrará unas opciones u otras dependiendo del rol del empleado.</li></ul>		

Tabla 3.3.2.1.4 Historia de usuario Diseño Menú Lateral.

Nombre	Funcionalidad Menú Lateral	ID	TFG-14
Sprint	1	Épica	Menú Lateral
Fecha creación	05/Feb/2016	Fecha realización	10/Mar/2016
Descripción	<ul style="list-style-type: none"><li>Cambiar de vista según ítem seleccionado.</li></ul>		

Tabla 3.3.2.1.5 Historia de usuario Funcionalidad Menú Lateral.

<b>Nombre</b>	Diseño Panel Principal	<b>ID</b>	TFG-18
<b>Sprint</b>	1	<b>Épica</b>	Panel Principal
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Generar vista de diseño de Panel Principal, que incluye vista de los últimos tres registros de horas, mensajes y recursos</li> <li>• Botón añadir al final de cada título, que te lleve a la vista correspondiente</li> </ul>		

Tabla 3.3.2.1.6 Historia de usuario Diseño Panel Principal.

<b>Nombre</b>	Funcionalidad Panel Principal	<b>ID</b>	TFG-19
<b>Sprint</b>	1	<b>Épica</b>	Panel Principal
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	13/May/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Muestra los últimos tres registros de horas, mensajes y recursos en una tabla. Si no existe ningún registro, mostrar "No existe ningún registro".</li> <li>• Cuando pulsamos encima de cada ítem que te lleve a la vista seleccionada.</li> </ul>		

Tabla 3.3.2.1.7 Historia de usuario Funcionalidad Panel Principal.

<b>Nombre</b>	Diseño Registro de Horas	<b>ID</b>	TFG-25
<b>Sprint</b>	1	<b>Épica</b>	Registro de Horas
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Un calendario, en cuyos días puede aparecer un indicador para saber si has imputado alguna hora.</li> <li>• Un botón "Borrar Mes" que elimina todas las imputaciones del mes.</li> <li>• Un botón "Ir a hoy", que te lleva al día actual</li> <li>• Un listado con los proyectos y sus horas correspondientes imputados en el día seleccionado o un mensaje "No tienes ninguna hora imputada para el día seleccionado" en el caso de no tener ninguna hora imputada (En cada celda dos botones, uno para editar la imputación y otro para borrarla).</li> <li>• Un botón "Añadir" para añadir una nueva imputación.</li> </ul>		

Tabla 3.3.2.1.8 Historia de usuario Diseño Registro de Horas.

<b>Nombre</b>	Funcionalidad Registro de Horas	<b>ID</b>	TFG-26
<b>Sprint</b>	1	<b>Épica</b>	Registro de Horas
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Mostrar un listado con las horas imputadas dependiendo del día seleccionado.</li> <li>Mostrar la vista "Añadir/Editar Imputación" cuando pulsas el botón "Editar".</li> <li>Mostrar pop-up de confirmación si pulsas botón "Eliminar".</li> <li>Mostrar la vista "Añadir/Editar Imputación" cuando pulsas el botón "Añadir".</li> </ul>		

Tabla 3.3.2.1.9 Historia de usuario Funcionalidad Registro de Horas.

<b>Nombre</b>	Diseño Editar Añadir Imputación	<b>ID</b>	TFG-27
<b>Sprint</b>	1	<b>Épica</b>	Registro de Horas
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Label "Editar datos del día: ".</li> <li>Label "Proyecto: " y un Select.</li> <li>Label "Horas: " y un TextEdit.</li> <li>Label "Razón: " y un Select.</li> <li>Label "Descripción: " y un TextEdit.</li> <li>Botón "Descartar Cambios".</li> <li>Botón "Guardar cambios".</li> </ul>		

Tabla 3.3.2.1.10 Historia de usuario Diseño Añadir Imputación.

<b>Nombre</b>	Modificación peticiones backend	<b>ID</b>	TFG-88
<b>Sprint</b>	1	<b>Épica</b>	MVC
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Modificar API backend para devolver los últimos tres registros, mensajes y facturas del usuario.</li> </ul>		

Tabla 3.3.2.1.11 Historia de usuario Modificación peticiones backend.

<b>Nombre</b>	Funcionalidad MVC	<b>ID</b>	TFG-92
<b>Sprint</b>	1	<b>Épica</b>	MVC
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016

Tabla 3.3.2.1.12 Historia de usuario Funcionalidad MVC.

### 3.3.2.2 Sprint 2

<b>Nombre</b>	Funcionalidad Editar/Añadir Imputación	<b>ID</b>	TFG-28
<b>Sprint</b>	2	<b>Épica</b>	Registro de Horas
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Mostrar en el Select de Proyectos los proyectos disponibles.</li> <li>• Mostrar en el Select de Razón las razones disponibles.</li> <li>• Volver a la vista "Registro Horas" si pulsamos el botón "Descartar Cambios" sin guardar ningún cambio.</li> <li>• Volver a la vista "Registro Horas" si pulsamos el botón "Guardar Cambios" modificando la imputación seleccionada con los valores que hemos modificado.</li> </ul>		

Tabla 3.3.2.2.1 Historia de usuario Funcionalidad Editar/Añadir Imputación.

<b>Nombre</b>	Diseño Recursos	<b>ID</b>	TFG-47
<b>Sprint</b>	2	<b>Épica</b>	Recursos
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Muestra el listado de recursos del grupo con el fondo en negro con un botón de "Añadir" a la derecha de este.</li> <li>• Muestra listado de los subrecursos del grupo específico con el fondo blanco.</li> <li>• Muestra una leyenda abajo de los tipos de documento que pueden crearse.</li> </ul>		

Tabla 3.3.2.2.2 Historia de usuario Diseño Recursos.

<b>Nombre</b>	Funcionalidad Recursos	<b>ID</b>	TFG-48
<b>Sprint</b>	2	<b>Épica</b>	Recursos
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Pulsando encima del recurso de grupo muestra o oculta el sublistado de recursos.</li> <li>• Pulsando encima del recurso específico muestra los datos del recurso para poder editarlo o verlo.</li> <li>• Pulsando encima de añadir recurso muestra un formulario en blanco donde poder insertar los datos del nuevo recurso</li> </ul>		

Tabla 3.3.2.2.3 Historia de usuario Funcionalidad Recursos.



<b>Nombre</b>	Diseño Crear/Ver/Editar Recurso	<b>ID</b>	TFG-49
<b>Sprint</b>	2	<b>Épica</b>	Recursos
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Muestra formulario con el título, Url, username, password, comentario, n_resource_type (Select), n_resource_group (Select).</li> <li>Muestra botones: <ul style="list-style-type: none"> <li>Vista editar: Cancel, Delete y Submit.</li> <li>Vista crear: Cancel y Submit.</li> </ul> </li> </ul>		

**Tabla 3.3.2.2.4 Historia de usuario Diseño Crear/Ver/Editar Recurso.**

<b>Nombre</b>	Funcionalidad Crear/Ver/Editar Recurso	<b>ID</b>	TFG-50
<b>Sprint</b>	2	<b>Épica</b>	Recursos
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Pulsando Cancel vuelve atrás sin guardar los cambios.</li> <li>Pulsando Delete elimina el recurso seleccionado.</li> <li>Pulsando Submit crea/modifica el recurso seleccionado.</li> <li>Campos requeridos: Título, n_resource_type, n_resource_group.</li> </ul>		

**Tabla 3.3.2.2.5 Historia de usuario Funcionalidad Crear/Ver/Editar Recurso.**

<b>Nombre</b>	Diseño Mi Perfil	<b>ID</b>	TFG-66
<b>Sprint</b>	2	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Generar vista de diseño de Mi Perfil. Incluye la foto del usuario, nombre, apellidos y mail.</li> <li>Botones de Editar perfil, notificaciones, acerca de y log out (ayuda-wizard como ampliación)</li> </ul>		

**Tabla 3.3.2.2.6 Historia de usuario Diseño Mi Perfil.**

<b>Nombre</b>	Funcionalidad Mi Perfil	<b>ID</b>	TFG-67
<b>Sprint</b>	2	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Muestra el listado de botones. Pulsando encima de cada ítem nos lleva a la vista seleccionada</li> </ul>		

**Tabla 3.3.2.2.7 Historia de usuario Funcionalidad Mi Perfil.**

<b>Nombre</b>	Diseño Editar Perfil	<b>ID</b>	TFG-68
<b>Sprint</b>	2	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Muestra formulario con los datos del usuario.</li> <li>• Botón de Cancelar y Submit.</li> <li>• Botón de volver atrás arriba a la izquierda.</li> </ul>		

**Tabla 3.3.2.2.8 Historia de usuario Diseño Editar Perfil.**

<b>Nombre</b>	Funcionalidad Editar Perfil	<b>ID</b>	TFG-69
<b>Sprint</b>	2	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Permite editar todos los campos del registro.</li> <li>• Cambiar contraseña.</li> <li>• Pulsando cancelar o atrás vuelve a la vista de Mi perfil.</li> <li>• Pulsando a Submit realiza llamada a la API para actualizar los datos del usuario.</li> <li>• Si algún campo está mal introducido no permite realizar la opción de submit.</li> <li>• Pulsando en la imagen del usuario permite modificarla por una existente en la galería del usuario o tomar una nueva de la cámara.</li> </ul>		

**Tabla 3.3.2.2.9 Historia de usuario Funcionalidad Editar Perfil.**

<b>Nombre</b>	Diseño Configurar Notificaciones	<b>ID</b>	TFG-70
<b>Sprint</b>	2	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Muestra dos seleccionables para push y correo.</li> <li>• Botón atrás arriba a la izquierda.</li> </ul>		

**Tabla 3.3.2.2.10 Historia de usuario Diseño configurar Notificaciones.**

<b>Nombre</b>	Funcionalidad Configuración Notificaciones	<b>ID</b>	TFG-71
<b>Sprint</b>	2	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Pulsando en el seleccionable se activa/desactiva las notificaciones push o correo. Realiza la llamada a la API para modificar las notificaciones en la sesión del usuario.</li> <li>• Pulsando atrás vuelve a la vista Mi Perfil.</li> </ul>		

**Tabla 3.3.2.2.11 Historia de usuario Funcionalidad Configuración Notificaciones.**

<b>Nombre</b>	Diseño Acerca de	<b>ID</b>	TFG-72
<b>Sprint</b>	2	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Muestra el logo de la empresa, nombre de la empresa, versión de la app y una breve descripción.</li> <li>Botón de atrás arriba a la izquierda.</li> </ul>		

Tabla 3.3.2.2.12 Historia de usuario Diseño Acerca de.

<b>Nombre</b>	Funcionalidad Acerca de	<b>ID</b>	TFG-73
<b>Sprint</b>	2	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	10/Mar/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Pulsando el botón de atrás nos lleva a la vista de Mi perfil.</li> </ul>		

Tabla 3.3.2.2.13 Historia de usuario Funcionalidad Acerca de.

### 3.3.2.3 Sprint 3

<b>Nombre</b>	Diseño Mensajes iOS	<b>ID</b>	TFG-36
<b>Sprint</b>	3	<b>Épica</b>	Mensajes
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Dos ItemButton en el Navigation Bar, uno para enviar un mensaje nuevo y otro para eliminar los mensajes seleccionados.</li> <li>SearchBar para filtrar los mensajes.</li> <li>Lista con los mensajes de salida o de entrada, en cuyas celdas muestra los campos From, To, Asunto, Descripción y Hora/día. Las celdas tienen la posibilidad de marcar alguna para ser eliminadas. Si hacemos sweep a la izquierda nos da la posibilidad de borrar.</li> <li>TabBar para mostrar la bandeja de salida o la de entrada.</li> </ul>		

Tabla 3.3.2.3.1 Historia de usuario Diseño Mensajes.

<b>Nombre</b>	Funcionalidad Mensajes	<b>ID</b>	TFG-37
<b>Sprint</b>	3	<b>Épica</b>	Mensajes
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Si pulsamos el ItemButton de enviar mensaje nos abre la vista "Mostrar/Redactar Mensaje" con los campos vacíos.</li> <li>• iOS: Si pulsamos el ItemButton de borrar nos muestra una confirmación para borrar los mensajes seleccionados. En el caso de no haber ningún mensaje seleccionado nos muestra un aviso.</li> <li>• Android: Si pulsamos y mantenemos en un mensaje nos lo selecciona dándonos la posibilidad de seleccionar más y apareciendo arriba una papelera (dónde estaba el icono de lupa de buscar) y la posibilidad de eliminarlo.</li> <li>• El SearchBar filtra los mensajes con la palabra clave que hayamos introducido.</li> <li>• Si pulsamos una celda, nos muestra la vista "Mostrar/Redactar Mensaje" con los datos del mensaje seleccionado.</li> <li>• Si hacemos sweep a la izquierda sobre una celda nos muestra la opción de borrar ese mensaje y nos muestra un mensaje de confirmación.</li> <li>• Dependiendo del Tab que tengamos seleccionado nos actualiza la tabla con los mensajes que tengan nuestro usuario en From o en To.</li> </ul>		

**Tabla 3.3.2.3.2 Historia de usuario Funcionalidad Mensajes.**

<b>Nombre</b>	Diseño Vista Mensajes	<b>ID</b>	TFG-38
<b>Sprint</b>	3	<b>Épica</b>	Mensajes
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Botón de atrás arriba a la izquierda.</li> <li>• Botón de eliminar y responder arriba a la derecha.</li> <li>• Título del mensaje.</li> <li>• From user.</li> <li>• Cuerpo del mensaje.</li> </ul>		

**Tabla 3.3.2.3.3 Historia de usuario Diseño Vista Mensajes.**

<b>Nombre</b>	Funcionalidad Vista Mensaje	<b>ID</b>	TFG-39
<b>Sprint</b>	3	<b>Épica</b>	Mensajes
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Pulsando atrás vuelve a la home de Mensajes.</li> <li>• Pulsando en eliminar nos muestra popup de confirmación.</li> <li>• Pulsando en responder nos muestra la vista de Responder mensaje.</li> </ul>		

**Tabla 3.3.2.3.4 Historia de usuario Funcionalidad Vista Mensaje.**

<b>Nombre</b>	Diseño Nuevo/Responder Mensaje	<b>ID</b>	TFG-40
<b>Sprint</b>	3	<b>Épica</b>	Mensajes
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Campo to: vacío en nuevo mensaje, relleno en responder mensaje.</li> <li>• Título: vacío en nuevo, RE:&lt;título mensaje anterior&gt; en responder mensaje.</li> <li>• Cuerpo de mensaje.</li> <li>• Botón de enviar.</li> </ul>		

Tabla 3.3.2.3.5 Historia de usuario Diseño Nuevo/Responder Mensaje.

<b>Nombre</b>	Funcionalidad Nuevo/Responder Mensaje	<b>ID</b>	TFG-41
<b>Sprint</b>	3	<b>Épica</b>	Mensajes
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Campos to, título y cuerpo obligatorios.</li> <li>• Pulsando en enviar mensaje se enviará el mensaje.</li> </ul>		

Tabla 3.3.2.3.6 Historia de usuario Funcionalidad Nuevo/Responder Mensaje.

<b>Nombre</b>	Diseño Cómo llegar	<b>ID</b>	TFG-54
<b>Sprint</b>	3	<b>Épica</b>	Cómo llegar
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Muestra el listado con todas las empresas de la base de datos (dirección, contacto y teléfono de contacto).</li> </ul>		

Tabla 3.3.2.3.7 Historia de usuario Diseño Cómo llegar.

<b>Nombre</b>	Funcionalidad Cómo llegar	<b>ID</b>	TFG-55
<b>Sprint</b>	3	<b>Épica</b>	Cómo llegar
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>• Pulsando encima de la dirección abre popup con los navegadores que tenga el usuario por defecto instalados en el dispositivo. Se pasan por parámetro la longitud y latitud (o dirección) de la empresa seleccionada.</li> <li>• Pulsando sobre el teléfono de contacto abre popup para confirmar la llamada de teléfono.</li> </ul>		

Tabla 3.3.2.3.8 Historia de usuario Funcionalidad Cómo llegar.

<b>Nombre</b>	Que los recursos aparezcan en función de los usuarios	<b>ID</b>	TFG-95
<b>Sprint</b>	3	<b>Épica</b>	Recursos
<b>Fecha creación</b>	10/Mar/2016	<b>Fecha realización</b>	04/Abr/2016

Tabla 3.3.2.3.9 Historia de usuario Que los recursos aparezcan en función de los usuarios.

<b>Nombre</b>	Traducir toda la app (i18n)	<b>ID</b>	TFG-98
<b>Sprint</b>	3	<b>Épica</b>	Not Specified
<b>Fecha creación</b>	10/Mar/2016	<b>Fecha realización</b>	04/Abr/2016

Tabla 3.3.2.3.10 Historia de usuario Traducir toda la app.

<b>Nombre</b>	Meter versión de la app en "About"	<b>ID</b>	TFG-102
<b>Sprint</b>	3	<b>Épica</b>	Not Specified
<b>Fecha creación</b>	10/Mar/2016	<b>Fecha realización</b>	04/Abr/2016

Tabla 3.3.2.3.11 Historia de usuario Meter versión de la app en "About".

<b>Nombre</b>	La funcionalidad "Set Date" que no aparezca si vienes del calendario	<b>ID</b>	TFG-104
<b>Sprint</b>	3	<b>Épica</b>	Not Specified
<b>Fecha creación</b>	10/Mar/2016	<b>Fecha realización</b>	04/Abr/2016

Tabla 3.3.2.3.12 Historia de usuario La funcionalidad "Set Date" que no aparezca si vienes del calendario.

<b>Nombre</b>	Cambiar foto de mi perfil	<b>ID</b>	TFG-107
<b>Sprint</b>	3	<b>Épica</b>	Ajustes/Mi Perfil
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	04/Abr/2016

Tabla 3.3.2.3.13 Historia de usuario Cambiar foto de mi perfil.

### 3.3.2.4 Sprint 4

<b>Nombre</b>	Notificaciones Backend	<b>ID</b>	TFG-63
<b>Sprint</b>	4	<b>Épica</b>	Not Specified
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	13/May/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Gestión de notificaciones para el usuario seleccionado en la base de datos (contemplar si se requiere notificación por push, correo, ambas o ninguna).</li> </ul>		

Tabla 3.3.2.4.1 Historia de usuario Notificaciones Backend.

Nombre	Funcionalidad Configuración Notificaciones	ID	TFG-71
Sprint	4	Épica	Ajustes/Mi Perfil
Fecha creación	05/Feb/2016	Fecha realización	13/May/2016
Descripción	<ul style="list-style-type: none"> <li>Pulsando en el seleccionable se activa/desactiva las notificaciones push o correo. Realiza la llamada a la API para modificar las notificaciones en la sesión del usuario.</li> <li>Pulsando atrás vuelve a la vista Mi Perfil.</li> </ul>		

Tabla 3.3.2.4.2 Historia de usuario Funcionalidad Configuración Notificaciones.

Nombre	Diseño de la vista del wearable	ID	TFG-86
Sprint	4	Épica	Wearable
Fecha creación	05/Feb/2016	Fecha realización	13/May/2016
Descripción	<ul style="list-style-type: none"> <li>Notificación de nuevo mensaje/recurso.</li> <li>Sweep a la izquierda para responder mensaje con texto predefinido, borrar, abrir app en el dispositivo y cancelar notificaciones.</li> </ul>		

Tabla 3.3.2.4.3 Historia de usuario Diseño de la vista del wearable.

Nombre	Funcionalidad wearable	ID	TFG-87
Sprint	4	Épica	Wearable
Fecha creación	05/Feb/2016	Fecha realización	13/May/2016
Descripción	<ul style="list-style-type: none"> <li>Pulsando en el mensaje/recurso posibilidad de ver el cuerpo del mensaje.</li> </ul>		

Tabla 3.3.2.4.4 Historia de usuario Funcionalidad wearable.

Nombre	Modificación petición mensajes Backend	ID	TFG-89
Sprint	4	Épica	Not Specified
Fecha creación	05/Feb/2016	Fecha realización	13/May/2016
Descripción	<ul style="list-style-type: none"> <li>Mirar la API para modificar la petición de devolución de mensajes. Si no se devuelven los mensajes en los que el usuario es el emisor, devolverlos.</li> </ul>		

Tabla 3.3.2.4.5 Historia de usuario Modificación petición mensajes Backend.

<b>Nombre</b>	Notificaciones Wearable Backend	<b>ID</b>	TFG-90
<b>Sprint</b>	4	<b>Épica</b>	Wearable
<b>Fecha creación</b>	05/Feb/2016	<b>Fecha realización</b>	13/May/2016
<b>Descripción</b>	<ul style="list-style-type: none"> <li>Investigación si se tiene que realizar alguna modificación en el backend y realizarla.</li> </ul>		

Tabla 3.3.2.4.6 Historia de usuario Notificaciones Wearable Backend.

### 3.3.2.5 Sprint 5

El sprint 5 se dedicó casi íntegramente a revivir historias de usuario incompletas o que sufrieron alguna modificación sobre la marcha.

En concreto, se volvieron a activar las historias de usuario con ID: TFG-19, TFG-63, TFG-86, TFG-87, TFG-90 y TFG-107.

Se añadió también una nueva:

<b>Nombre</b>	Hacer capturas de pantalla de las dos plataformas	<b>ID</b>	TFG-109
<b>Sprint</b>	5	<b>Épica</b>	Not Specified
<b>Fecha creación</b>	04/Abr/2016	<b>Fecha realización</b>	13/May/2016

Tabla 3.3.2.5.1 Historia de usuario Hacer capturas de pantalla de las dos plataformas.

### 3.3.2.6 Sprint 6

<b>Nombre</b>	Memoria iOS	<b>ID</b>	TFG-113
<b>Sprint</b>	6	<b>Épica</b>	Memorias
<b>Fecha creación</b>	13/May/2016	<b>Fecha realización</b>	20/May/2016

Tabla 3.3.2.6.1 Historia de usuario Memoria iOS.

<b>Nombre</b>	Bugs iOS	<b>ID</b>	TFG-116
<b>Sprint</b>	6	<b>Épica</b>	Revisión App
<b>Fecha creación</b>	13/May/2016	<b>Fecha realización</b>	20/May/2016

Tabla 3.3.2.6.2 Historia de usuario Bugs iOS.

<b>Nombre</b>	Diseño iOS	<b>ID</b>	TFG-118
<b>Sprint</b>	6	<b>Épica</b>	Revisión App
<b>Fecha creación</b>	13/May/2016	<b>Fecha realización</b>	20/May/2016

Tabla 3.3.2.6.3 Historia de usuario Diseño iOS.



## 4 Desarrollo

En el cuarto capítulo de esta memoria se hablará de forma más detallada acerca del desarrollo que se ha llevado a cabo para la realización de la aplicación. También se explicará cómo se ha organizado el proyecto, es decir, la distribución de las clases de programación, así como los recursos necesarios y cómo se han ido realizando las diferentes pruebas de calidad de la app.

### 4.1 Estructura del proyecto

Cualquier proyecto de iOS que se inicie se organiza en carpetas en Xcode. Esta organización puede coincidir o no con la organización de las carpetas en el directorio del ordenador donde son almacenadas. Lo ideal, y lo que dictan las buenas prácticas, es que ambas organizaciones sean iguales, ya que será más fácil acceder a una clase concreta del proyecto de esta manera. En mi caso, y siguiendo las buenas prácticas que dicta SCRUM, lo he organizado de esa forma, aunque a continuación solo mostraré la organización que he seguido en Xcode:

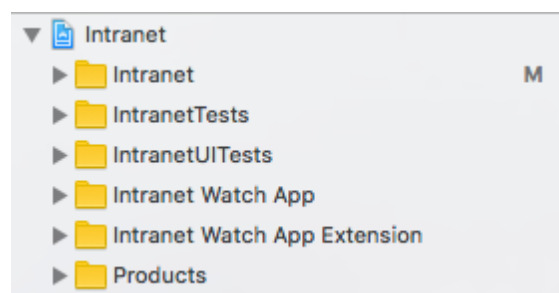


Figura 4.1.1 Organización de las carpetas del proyecto.

Todo proyecto de Xcode tiene una carpeta, denominada *NombreDelProyecto.xcodeproj* que engloba al resto de carpetas y archivos necesarios para la compilación del mismo. En la imagen se ve representada por la carpeta superior de color azul denominada Intranet. En Xcode no se muestra la extensión del archivo.

Automáticamente, al crear un nuevo proyecto, el programa genera 4 carpetas: *NombreDelProyecto*, *NombreDelProyectoTests*, *NombreDelProyectoUITests* y *Products*. En mi caso se han generado dos carpetas más, que contendrán los archivos necesarios para la inclusión del Apple Watch: *NombreDelProyecto Watch App* y *NombreDelProyecto Watch App Extension*.

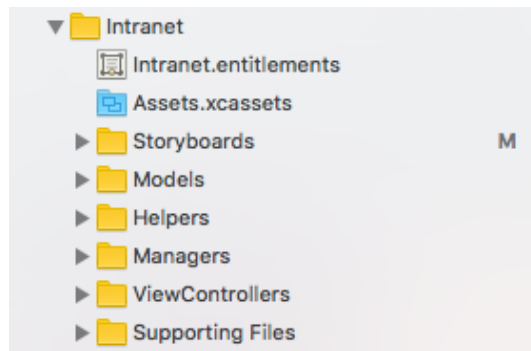


Figura 4.1.2 Organización de las carpetas de la aplicación móvil.

La carpeta *Intranet* engloba los archivos y carpetas necesarios para la aplicación móvil. *Intranet.entitlements* contiene datos de la aplicación wearable; la carpeta *Assets.xcassets* contiene las imágenes de la aplicación; la carpeta *Storyboards* contiene todos los Storyboards que rigen el diseño; *Models* contiene todos los modelos; en *Helpers* encontramos clases que nos serán útiles a lo largo del desarrollo; la carpeta *Managers* contiene todas aquellas clases que van a comunicar la aplicación con terceros; en la carpeta *ViewControllers* se encuentran todas las clases encargadas de la función Controlador del MVC, mencionado en el apartado de Estructura de iOS (Terminología); y, por último, la carpeta *Supporting Files* contiene archivos necesarios para el proyecto.

A continuación explicaré el contenido de todas las carpetas mencionadas:

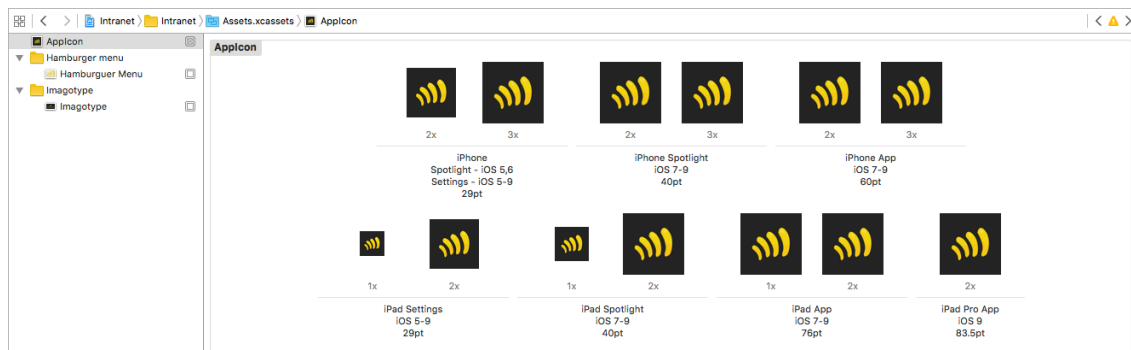


Figura 4.1.3 Diferentes tamaños de imágenes para diferentes dispositivos.

La carpeta *Assets.xcassets* contiene las imágenes del proyecto en las diferentes medidas necesarias para todos los dispositivos Apple y, por tanto, para todos los tamaños de pantalla. La aplicación selecciona automáticamente la imagen con el tamaño apropiado dependiendo del dispositivo en el que sea instalada.

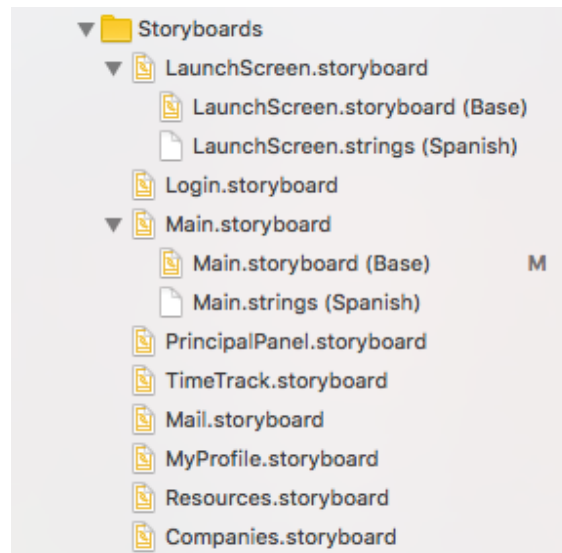


Figura 4.1.4 Organización de los Storyboards de la aplicación móvil.

En la carpeta *Storyboards* vemos todos y cada uno de los Storyboards de la aplicación. El primero de ellos contiene el diseño de la pantalla que aparece cuando abres la aplicación. Después, hay un Storyboard por cada módulo del proyecto:

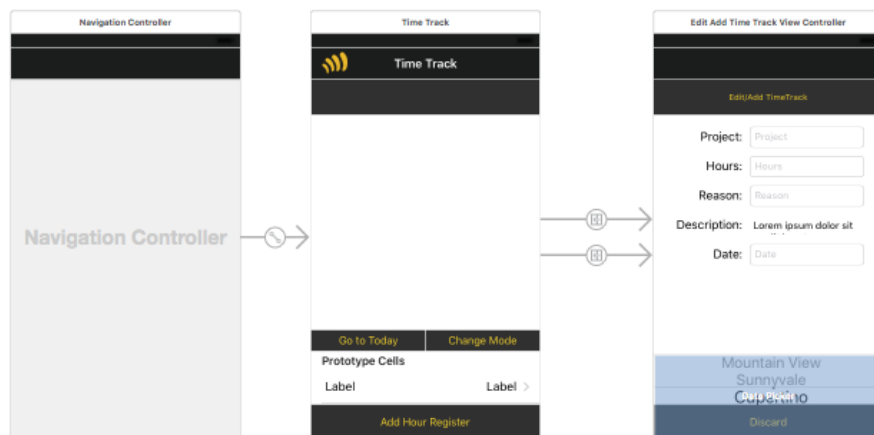


Figura 4.1.5 Ejemplo de Storyboard de la aplicación móvil.

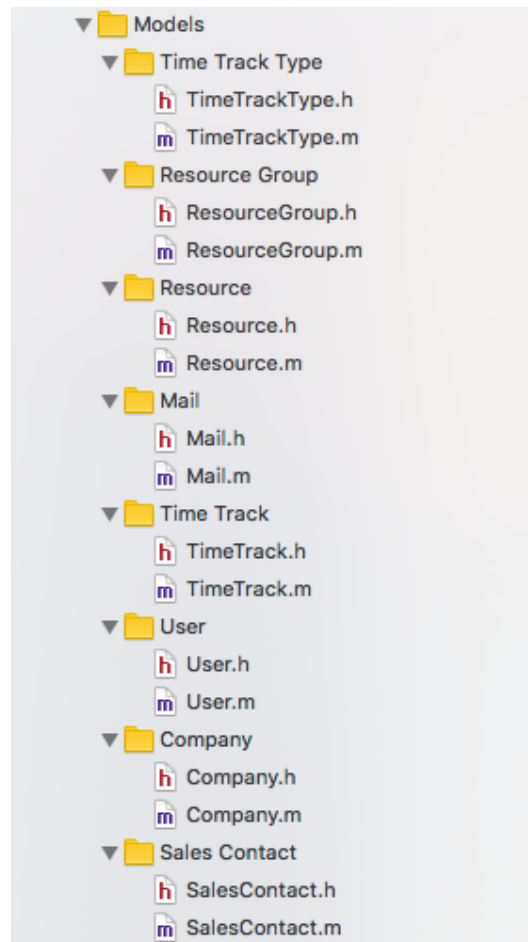


Figura 4.1.6 Organización de la carpeta de modelos de la aplicación móvil.

En la carpeta *Models* vemos 8 modelos diferentes. En cada uno de ellos se va a almacenar la información de un objeto, es decir, tendrá tantos argumentos como columnas existan en su tabla de la base de datos:

```

@implementation User
- (id)initWithActive: (BOOL) active
    AndCreated: (NSDate *) created
    AndEmail: (NSString *) email
    AndFirstName: (NSString *) firstname
    AndFK_K_Role: (int) fk_k_role
    AndFK_N_Company: (int) fk_n_company
    AndId: (int) _id
    AndLastName: (NSString *) lastname
    AndNif: (NSString *) nif
    AndPassword: (NSString *) password
    AndUrl_Pic: (NSURL *) url_pic
    AndDeviceId: (NSString *) device_id
    AndOS: (NSString *) OS
{
    self = [super init];
    if (self) {
        self.active = active;
        self.created = created;
        self.email = email;
        self.firstname = firstname;
        self.fk_k_role = fk_k_role;
        self.fk_n_company = fk_n_company;
        self._id = _id;
        self.lastname = lastname;
        self.nif = nif;
        self.password = password;
        self.url_pic = url_pic;
        self.device_id = device_id;
        self.OS = OS;
    }
    return self;
}
@end
                
```

Campo
id
fk_k_role
password
firstname
lastname
email
nif
url_pic
fk_n_company
active
employee
fk_k_users
created
holidays
device_id
os

Figura 4.1.7 Ejemplo de modelo *User*. Campos de la tabla *user* de la base de datos.

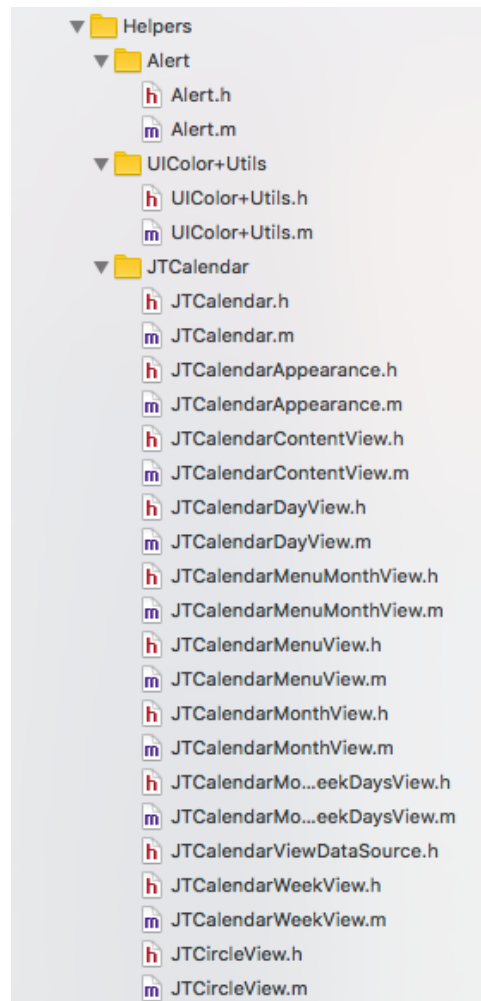


Figura 4.1.8 Organización de la carpeta *Helpers* de la aplicación móvil.

En la carpeta *Helpers* hay tres carpetas. La primera contiene los métodos necesarios para que aparezca una alerta; en la segunda encontramos las clases que contienen los colores prefijados que queremos utilizar; y en la tercera aparecen los métodos necesarios para el diseño y la funcionalidad del calendario del módulo de Registro de horas.

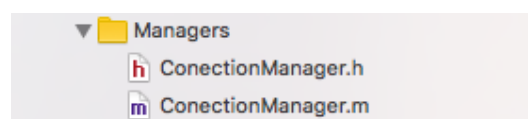


Figura 4.1.9 Organización de carpeta *Managers* de la aplicación móvil.

Solo he implementado un Mánager para el desarrollo de esta aplicación, y se ubica en la carpeta *Managers*. Es el Mánager encargado de las peticiones al Backend.

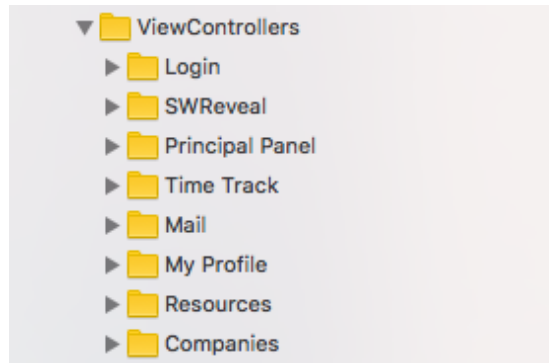


Figura 4.1.10 Organización de la carpeta *ViewControllers* de la aplicación móvil.

El siguiente directorio importante es el de *ViewControllers*. En él se encuentran todas las clases que se encargan de las diferentes funcionalidades que tiene que llevar a cabo cada vista.

El nombre de cada controlador será el nombre de la vista que representan seguido de los sufijos *ViewController* o *TableViewController*, según se vayan a encargar de la funcionalidad de un View o un TableView respectivamente.

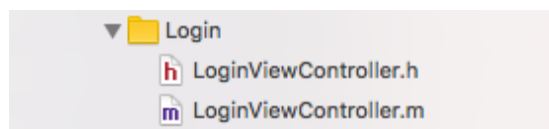


Figura 4.1.11 Carpeta Login.

El Login consta de una vista y es un View, por lo cual será manejado por un único ViewController.

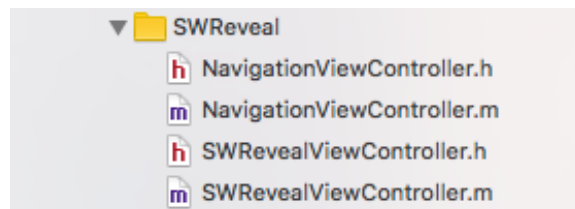


Figura 4.1.12 Carpeta SWReveal.

*SWReveal* es la carpeta que contiene las dos clases que se van a encargar del menú lateral y de posibilitar que aparezca desde todos los módulos.

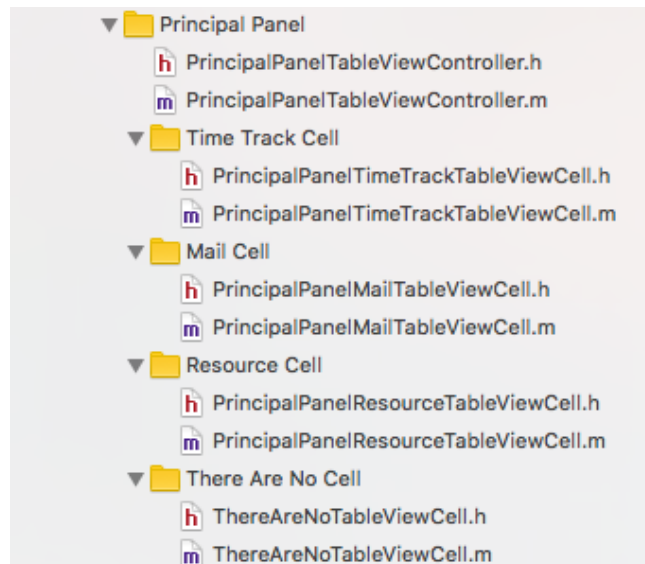


Figura 4.1.13 Carpeta *Principal Panel*.

El directorio *Principal Panel* comienza a ser más complejo que los anteriores. Contiene un controlador para el listado de los tres últimos recursos, mensajes o registros de horas; y un controlador por cada uno de los tipos de celda.

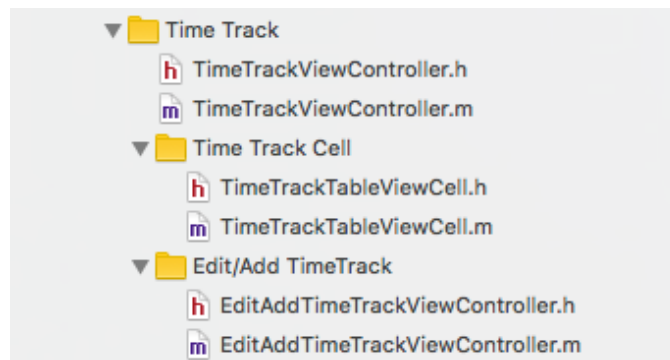


Figura 4.1.14 Carpeta *Time Track*.

El módulo *Time Track* consta de dos vistas: una en la que aparece el calendario, controlada por *TimeTrackViewController*; y otra en la que se puede editar, añadir o eliminar un registro, controlada por el controlador *EditAddTimeTrackViewController*.

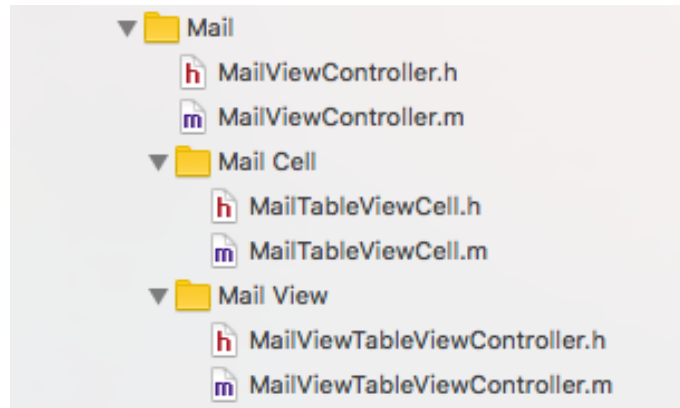


Figura 4.1.15 Carpeta *Mail*.

En el directorio *Mail* se ubican los controladores de la vista de las bandejas de entrada y salida; de las celdas de la tabla de las bandejas; y el de la vista de un correo.

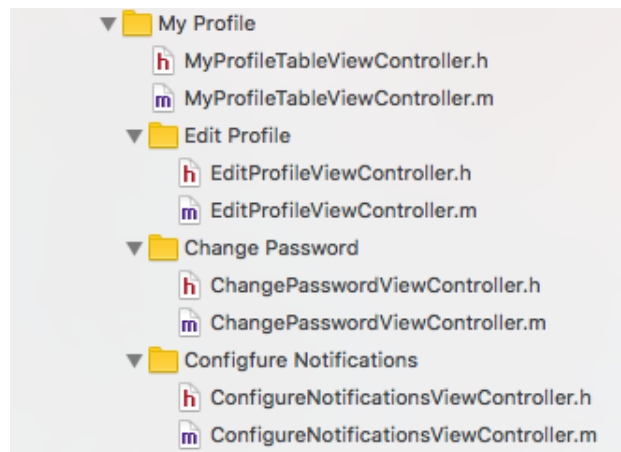


Figura 4.1.16 Carpeta *My Profile*.

La carpeta *My Profile* contiene los controladores para la vista del perfil, de editar perfil, de cambiar la contraseña y la de configurar notificaciones.



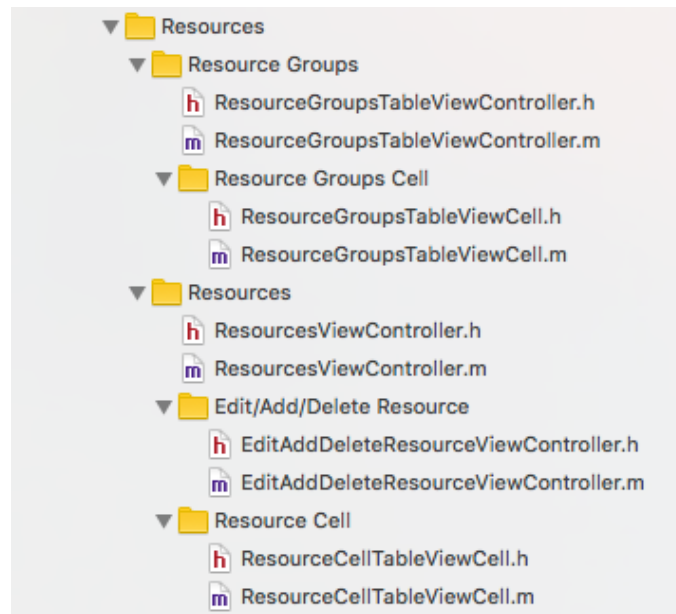


Figura 4.1.17 Carpeta Resources.

El módulo de *Resources* tiene tres pantallas, una que muestra los grupos de recursos, controlado por el primero de los controladores que aparecen en la imagen; otra en la que aparecen los recursos para un grupo seleccionado, controlado por el *ResourceViewController*; y la última en la que se puede editar, añadir o eliminar un recurso concreto, controlado por el último controlador, *EditAddDeleteResourceViewController*.

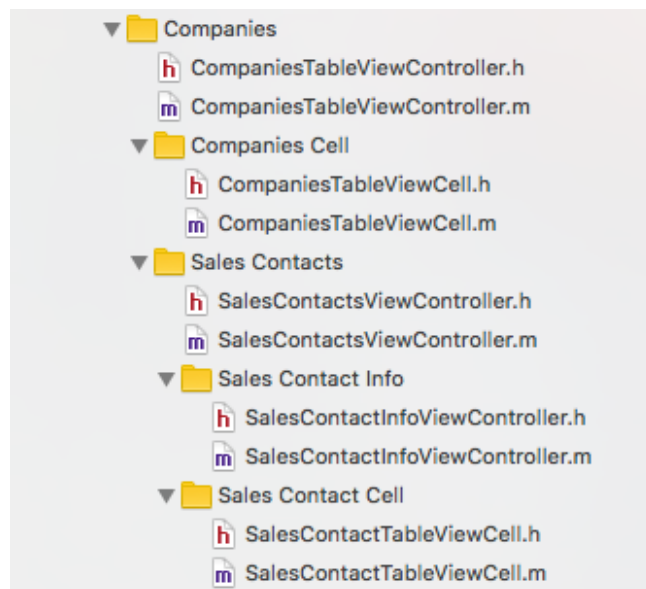


Figura 4.1.18 Carpeta Companies.

En la carpeta *Companies* aparecen los controladores de las vistas del listado de empresas con las que se está trabajando, del listado de los contactos de cada empresa y de la vista de la información de un contacto en concreto.

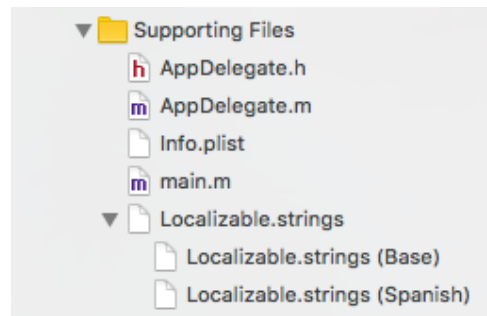


Figura 4.1.19 Carpeta *Supporting Files*.

En el último de los directorios incluidos en la carpeta *Intranet* es el de *Supporting Files*, uno de los que genera Xcode automáticamente al crear un nuevo proyecto. En él está el *AppDelegate*, que comprende las funciones encargadas del funcionamiento básico de la aplicación; el *Info.plist*, que contiene las *keys* requeridas por el sistema; el *main.m*, que contiene la función *main()*, encargada de iniciar la aplicación; y el archivo *Localizable.strings*, gracias al cual la app puede soportar varios idiomas.

## 4.2 Funcionalidades

### 4.2.1 ConectionManager

El ConectionManager es, como su propio nombre indica, un manager, es decir, un gestor. En este caso, se ocupa de la gestión de la conexión con la base de datos. Es el encargado de hacer las peticiones al backend y devolver los datos necesarios para pintarlos en las vistas.

Las peticiones son por *POST*, es decir, hay que pasar una serie de argumentos junto a la URL donde se alberga el backend. Estos argumentos son tres: ***service***, que contiene el nombre del servicio que se quiere lanzar, por ejemplo, *initSession*; ***auth***, o token, es el identificador propio de cada usuario que permite saber si un usuario está logueado, es un argumento necesario para todas las peticiones excepto el login; y ***args***, que son los diferentes argumentos que precisa el servicio, en el caso del *initSession* necesitará el email y el password.

La clase ConectionManager.m tiene implementadas una clase llamada *connect()* y una clase por cada servicio. La clase *connect()* recibe los argumentos necesarios para realizar la petición al backend y realiza la conexión, por ello cada clase de servicio recibe y construye los argumentos, que más tarde le pasará por parámetro.

El backend está programado en php y los datos que devuelve el ConectionManager tienen formato *json*. Si realizamos la petición de *initSession*, el *json* generado tendrá la siguiente estructura:

```

{
  "command": "ok",
  "args": {
    "code": "SHTMBd6lEbLVZZqcY0Q8pruw77wIdd0WKKxbNumh",
    "user": {
      "list": {
        "id": "13",
        "fk_k_role": "1",
        "password": "MTIzNA==",
        "firstname": "Guillermo",
        "lastname": "Blanco",
        "email": "guillermo.blanco@armadilloamarillo.com",
        "nif": "123123123B",
        "url_pic": "hola",
        "fk_n_company": "17",
        "active": "1",
        "created": "2016-03-23 10:33:23",
        "device_id": ""
      }
    }
  }
}

```

Si la conexión no se ha realizado con éxito el *json* mostrará un campo de *error*.

La aplicación podrá entonces manejar esos datos como lo prefiera, aunque lo más apropiado, y lo que hace la INTRANET, es crear un objeto con ellos, de tal forma que será más fácil su gestión.

#### 4.2.2 Login

- Cuando pulsamos el botón Login, se comprueba que los dos campos, usuario y contraseña, no están vacíos y se hace la petición de *initSession* al backend.
- Se almacenan los datos del usuario en la memoria de la aplicación, ya que son datos que se van a utilizar en distintas partes del desarrollo.

#### 4.2.3 Módulo Menú lateral

- Carga la foto de perfil, el nombre y los apellidos.
- Muestra todos los módulos de la aplicación y, pulsando en cualquiera de ellos, abre su respectiva vista.

#### 4.2.4 Módulo Panel Principal

- Realiza la petición al backend de *getPanelInfo* que devuelve los tres últimos registros de hora imputados, los tres últimos correos recibidos y los tres últimos recursos que se nos han asignado.
- Tiene tres botones “+” que abren las vistas de “nueva imputación”, “nuevo correo” o “nuevo recurso” para imputar un nuevo registro de horas, enviar un correo o añadir un nuevo recurso respectivamente.

#### 4.2.5 Módulo Registro de Horas

- Cuando se carga la vista, aparece en la parte superior de la pantalla un calendario que muestra un mes con el día de hoy seleccionado. Puedes seleccionar otro día y, debajo del mes, hay dos botones. Uno te lleva directamente al día de hoy y el otro cambia el modo del calendario a semana.
- Debajo del calendario hay una lista que muestra los registros de hora del día seleccionado tras realizar la petición de *getTimeTrackForDate*. Si no hay ninguna imputación ese día aparece un mensaje.
- Pulsando una imputación se abre una vista que nos muestra la información detallada de esta, con los campos proyecto, horas, razón y descripción. Existe la opción de borrarla o modificarla, realizando las peticiones de *deleteTimeTrack* o *updateTimeTrack* respectivamente.
- En la parte inferior de la pantalla hay un botón para añadir una nueva imputación en el día seleccionado. Para ello se lanza la petición con el servicio *addTimeTrack*.

#### 4.2.6 Módulo Correo

- Al cargar la vista muestra un listado con los correos de la bandeja de entrada tras la petición de *getMyMessages*. Cada celda tiene el nombre de la persona que ha enviado el correo, el asunto del correo, el mensaje y la fecha.
- En la parte superior hay un *Search Bar*, que es una barra que permite filtrar los correos por asunto, nombre de la persona que lo ha enviado y cualquier palabra del mensaje.
- En la parte inferior hay dos botones que permiten cambiar entre bandeja de entrada y bandeja de salida. Para mostrar los mensajes de la bandeja de salida se llama a *getSentMessages*.
- En la parte derecha del Navigation Bar hay un botón que permite enviar un nuevo correo. Para hacerlo se realiza la petición de *sendMessage*, pasándole por parámetro el usuario destino, el asunto y el mensaje.
- Si pulsamos un correo se abre una vista con la información detallada del correo. En esta ocasión en la parte derecha del Navigation Bar aparecen dos

botones, uno que permite responder el correo y otro que lo borra. Para la primera acción se utiliza la mencionada *sendMessage* y para la segunda *deleteMessage*.

#### 4.2.7 Módulo Mi perfil

- La vista que se carga muestra la foto de perfil del usuario, su nombre y apellidos y el correo, información que capta de la memoria interna de la aplicación.
- Debajo aparece una lista con posibles acciones que se pueden hacer.
- La primera es editar perfil. Una vez más se carga la fotografía de perfil. Si pulsamos encima, la app nos ofrece la posibilidad de cambiarla, seleccionando una imagen del carrete o abriendo la cámara y tomando una nueva foto. También podemos editar el nombre, el apellido, el email y el NIF. Los servicios que se utilizan en este caso son *updateProfileImage* y *updateUser* respectivamente. Para el primero, los parámetros que necesita son un *string* con la imagen encodeada en base64 y un nombre para almacenarla; para el segundo necesita todos los datos del usuario, es decir, el nombre, el apellido, el email y el NIF. A continuación se explicará cómo de ha llevado a cabo la integración de la cámara.
- La segunda opción es cambiar la contraseña. Como es habitual, se abre una vista en la cual tendremos que introducir la contraseña actual, confirmarla y escribir la nueva. La aplicación comprueba que la contraseña actual y la confirmación sean iguales y correspondan con el usuario en cuestión a través de la petición *updatePassword*, que recibe como parámetro la nueva contraseña.
- La tercera celda es la de configurar notificaciones. Si la pulsamos se abre una vista que muestra dos posibles ajustes: las notificaciones push y las notificaciones de correo con un switch en cada una de ellas. La primera acción se realiza llamando al servicio *updateDeviceID*, que lleva a cabo el registro con el sistema de notificaciones push de Apple APNS, que se explicará más adelante. La segunda no está implementada aún.
- Las dos siguientes opciones son: “Ayuda” y “Acerca de”.
- La última opción que podemos seleccionar es “Cerrar sesión”, que hace logout y muestra la ventana de login de nuevo. Llama al servicio *closeSession*.

#### 4.2.8 Módulo Recursos

- Lo primero que vemos al cargar esta vista es un listado con todos los grupos de recursos a los que tenemos acceso. El servicio que lleva a cabo esta petición es el *getResource\_groupForUser*.

- Si pulsamos un grupo de recursos nos lleva a una pantalla en la que aparecen los recursos disponibles para ese grupo en concreto, acción que es posible gracias al servicio *getResourcesForGroup*, al cual le pasas por parámetro el id del grupo de recurso que hemos abierto.
- En la parte inferior de la pantalla hay un botón que nos permite añadir un recurso nuevo. Se abre una ventana con un formulario para rellenar todos los campos de un recurso y se realiza la petición al backend a través del servicio *addResource*, pasándole por parámetro todos esos campos.
- Pulsando un recurso se abre una vista con la información concreta de ese recurso. Podemos borrarlo o editarlo con los servicios de *deleteResource*, pasándole por parámetro el id del recurso en cuestión, o *updateResource*, pasándole todos los campos de recurso, respectivamente.

#### 4.2.9 Módulo Empresas

- Al abrir la vista de empresas se carga en una lista todas las empresas con la que se está trabajando. Esto es gracias al servicio *getCompanies*, que devuelve dicho listado.
- Cuando pulsamos una, aparece un nuevo listado con los contactos de esa empresa en concreto. Esto es posible por la existencia de un servicio llamado *getSalesContactForCompany*, al cual hay que pasarle por parámetro el id de su empresa.
- En la parte inferior de la pantalla hay un botón llamado “Cómo llegar”. Si lo pulsamos nos aparece una alerta en la cual podemos seleccionar una aplicación de mapas que haya instalada en el dispositivo y abrirla con la dirección de la empresa que estamos consultando. La gestión de GPS se explicará más adelante.
- Si pulsamos un contacto en concreto se abre una vista con la información del contacto en cuestión y nos ofrece la posibilidad de llamar al número de teléfono o enviar un mail. Ambas integraciones se explicarán, una vez más, a continuación.

#### 4.2.10 APNs (Apple Push Notification service) [17]

APNs (Apple Push Notification service) es el servicio de notificaciones push que ofrece Apple para sus dispositivos.

Este servicio se basa en un servidor creado por el usuario, también conocido como proveedor, que recopila datos y es el encargado de enviar la notificación.

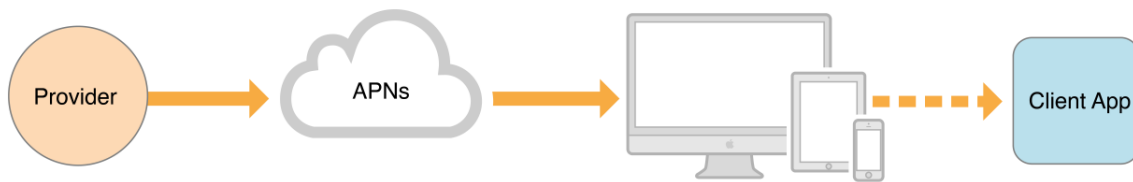


Figura 4.2.10.1 Trayecto de una notificación push<sup>17</sup>.

La figura muestra un esquema de la trayectoria que toma una notificación. Cuando el proveedor considera oportuno enviar una notificación, lo hace directamente al servidor APNs, junto con un *device token*. Después, los servidores APNs se encargan de enviar la notificación al dispositivo de usuario correcto y el sistema operativo entrega la notificación a la aplicación cliente.

El *device token* es un conjunto de números y letras que van ligados a un terminal concreto, por ejemplo “333507b5 855ba575 d26821fc 08857f0e 8d40c582 4f4b56ad 40e820ce a532db71”. Contiene información que permite a APNs localizar el dispositivo en el cual está instalada la aplicación cliente. Es suministrado por la aplicación tras realizar el registro en el servicio de notificaciones.

El *payload* es un *json* que contiene información acerca de la configuración de la notificación. Entre otros, puede contener el mensaje que ha de aparecer o el sonido de la alerta.

```
aps = {
  action = {
    identifier = firstButtonAction;
    title = "First Button";
  }
  alert = {
    body = "Notificaciones activadas";
    detail = "Buenas tardes";
    detail_user = "12";
  }
  badge = 1;
  category = myCategory;
  Notification_type = 1;
  Sound = "oven.caf";
};
```

<sup>17</sup> Apple Inc., “Figure 3-1 Pushing a remote Notification from a provider to a client app”, 2016, [en línea].

Recuperada de:

<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html> [consulta: 7 de junio de 2016]

Para realizar la conexión con el servidor APNs, tanto el proveedor como el dispositivo lo hacen a través de una autenticación TLS peer-to-peer, que es un protocolo criptográfico. APNs comprueba que los certificados de ambos son correctos y valida la conexión:

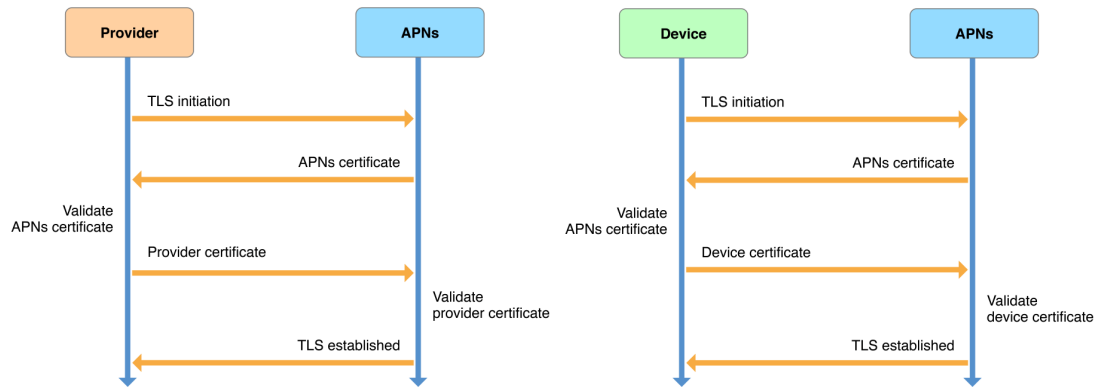


Figura 4.2.10.2 Esquema de autenticación TLS<sup>18</sup>.

Para generar el *token*, el dispositivo realiza la petición de registro y APNs devuelve el *token* generado a partir del certificado. Después este deberá pasárselo al proveedor.

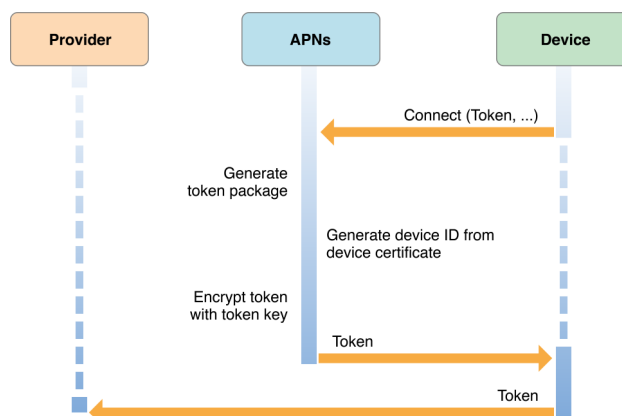


Figura 4.2.10.3 Esquema de generación de token<sup>19</sup>.

<sup>18</sup> Apple Inc., “Figure 3-3 Establishing connection trust between a provider and APNs”, 2016, [en línea].

Recuperada de:

<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationPG/Chapters/ApplePushService.html> [consulta: 7 de junio de 2016]

<sup>19</sup> Apple Inc., “Figure 3-5 Managing the device token”, 2016, [en línea]. Recuperada de:

<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationPG/Chapters/ApplePushService.html> [consulta: 7 de junio de 2016]



Por último, para enviar una notificación, el registro tiene que estar activo. Es entonces cuando el proveedor envía el *payload* junto con el *token* al APNs y este se encarga de entregárselo al dispositivo.

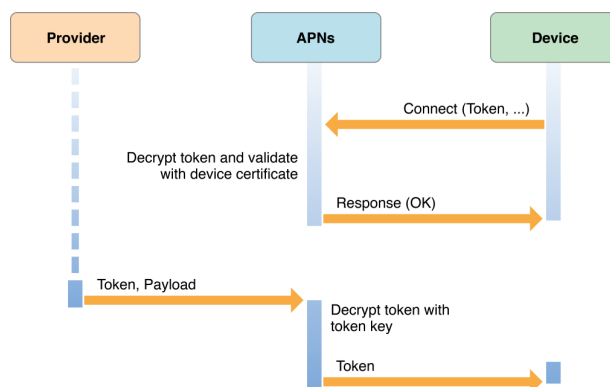


Figura 4.2.10.3 Esquema de envío de *payload*<sup>20</sup>.

#### 4.2.11 GPS

Para lanzar una aplicación de mapas, lo primero que hay que hacer es importar una librería llamada *MapKit/MapKit.h*.

Después es sencillo. En el caso de que queramos abrir la aplicación Maps integrada en el iPhone tendremos que implementar las siguientes líneas de código:

```

CLLocationCoordinate2D rdOfficeLocation =
CLLocationCoordinate2DMake(self.company.latitude, self.company.longitude);
MKPlacemark *placemark = [[MKPlacemark alloc]
initWithCoordinate:rdOfficeLocation addressDictionary:nil];
MKMapItem *item = [[MKMapItem alloc] initWithPlacemark:placemark];
item.name = self.company.name;
[item openInMapsWithOptions:nil];
  
```

Si lo que queremos es abrir Google Maps, debemos usar el siguiente código, el cual comprueba si la aplicación está instalada o no:

```

if ([[UIApplication sharedApplication] canOpenURL: [NSURL
URLWithString:@"comgooglemaps-x-callback://"]])
{
    [[UIApplication sharedApplication] openURL: [NSURL
URLWithString:[NSString stringWithFormat:@"comgooglemaps-x-
callback://?q=%@&center=%f,%f&zoom=16", companyName,
self.company.latitude, self.company.longitude]]];
} else
{
    NSLog(@"Can't use comgooglemaps://");
}
  
```

<sup>20</sup> Apple Inc., "Figure 3-7 Identifying a device using the device token", 2016, [en línea]. Recuperadas de: <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationPG/Chapters/ApplePushService.html> [consulta: 7 de junio de 2016]

Esta información viene perfectamente detallada en la página web de Google Developers [18]. Puedes introducir más parámetros en la búsqueda como el zoom del mapa cuando se abra.

En ambos casos hay que pasar por parámetro las coordenadas, altitud y longitud.

#### 4.2.12 Llamada

Para realizar una llamada de teléfono tendremos que lanzar la aplicación de teléfono integrada en el iPhone. Simplemente habrá que pasarle por parámetro el número de teléfono al que queremos llamar. Bastará con un par de líneas de código como las siguientes:

```
NSString *phoneNumber = [@"telprompt://"
stringByAppendingString:self.lbSalesContacPhone.text];
[[UIApplication sharedApplication] openURL:[NSURL URLWithString:phoneNumber]];
```

#### 4.2.13 Email

Enviar un email es algo más complejo. Debemos importar dos librerías llamadas *MessageUI/ MFMailComposeViewController* y *MessageUI/ MFMessageComposeViewController*.

Después bastará con indicar el correo electrónico destino, el asunto y el mensaje que queremos enviar para que se abra la aplicación de Mail del iPhone con estos campos rellenos.

#### 4.2.14 Backend

Algunos de los servicios empleados para hacer peticiones al backend ya estaban implementados, la minoría, pero había muchos que eran necesarios y, como tanto la aplicación iOS como la aplicación Android los compartirían, decidimos dividirnos el trabajo, mi compañero y yo, y que cada uno implementase ciertas funciones.

Las funciones que yo implementé se muestran en la siguiente tabla:

Servicios	Módulo	Descripción	
		Parámetros que se le pasan	Devuelve
deleteResource	Resource	token, id	"ok" o "error".
updateResource	Resource	token, id, title, url, username, password, comment, fk_n_resource_type y fk_n_resource_group	La estructura del recurso actualizado.
addResource	Resource	token, title, url, username, password, comment, fk_n_resource_type y fk_n_resource_group	La estructura del recurso añadido.
getResourcesForGroup	Resource	token, group	"warning" si no hay recursos para ese grupo o todos los recursos de ese grupo.
getMyMessages	Mail	token	Todos los mails en los que el usuario es el destinatario.
getSentMessages	Mail	token	Todos los mails en los que el usuario es el que lo envía.
sendMessage	Mail	token, to, subject, message, push_notification_message	"ok" o "error"
deleteMessage	Mail	token, message_id, option	"ok" o "error"
setMailRead	Mail	token, id	"ok" o "error"
sendIOSNotification	Notifications	token, device_id, message, detail, detail_user, notification_type	No devuelve nada, es una función llamada desde otros servicios.
getResource_groupForUser	Resource_group	token	Los grupos de recursos con los que está relacionado el usuario.
getCompanies	Company	token	Las empresas cliente.

Tabla 4.2.14.1 Funciones implementadas en el backend.

Mi compañero se ocupó de las funciones correspondientes al módulo de **Mi Perfil**, tales como las encargadas de modificar los datos de perfil de usuario, la contraseña y la imagen; al módulo de **Panel Principal**, cuya única llamada devuelve los tres últimos mensajes, recursos e imputaciones de horas; y el módulo de **Registro de Horas**, cuyas funciones gestionaban todos los registros de horas, es decir, devolver las imputaciones de un día en concreto, añadir una nueva, modificar una existente o eliminarla.

En cuanto a las funciones que ya estaban programadas cuando empezamos el desarrollo de la aplicación móvil, las más importantes eran la de *login* y la de *logout*. La primera, si las credenciales son correctas, crea una sesión y un *token* único a cada usuario para que el resto de llamadas a la base de datos se realicen de manera segura. La segunda hace lo contrario, es decir, elimina esa sesión iniciada y borra el *token* del usuario. El resto de funciones implementadas no las utilizamos porque devolvían, entre otros, todos los recursos, independientemente del usuario que lo solicite o del día, por lo cual no eran de interés para nuestras aplicaciones.

### 4.3 Pruebas de calidad

El QA (Quality Assurance o aseguramiento de la calidad) es el proceso que se sigue cuando se quiere medir los requisitos de calidad de un producto o un servicio.

En mi caso, el QA han sido una serie de pruebas unitarias que se han ido haciendo cuando se terminaba un módulo, por lo cual, las pruebas que se han realizado han sido las siguientes:

- Login (Superado)
  - Comprobar que el diseño esté bien.
  - Comprobar que funcionan los campos de texto y que el de password muestra asteriscos en vez de la contraseña.
  - Comprobar que al pulsar el botón login contempla errores de conexión, usuario y contraseña no coincidentes o campos vacíos.
  - Comprobar que al pulsar el botón login y ser correctas las credenciales hace login y muestra el Panel Principal.
- Menú Lateral (Superado)
  - Comprobar que el diseño esté bien.
  - Comprobar que se muestra desde cualquier modulo.
  - Comprobar que muestra la imagen y el nombre.
  - Comprobar que muestra todos los módulos cuando son pulsados.

- Panel Principal (Superado)
  - Comprobar que el diseño esté bien.
  - Comprobar que muestra los tres últimos registros de hora, correos y recursos.
  - Comprobar que si no existen algunos de los campos muestra un mensaje.
  - Comprobar que los botones “+” funcionan.
  - Comprobar que pulsando uno de los campos muestra información detallada.
  
- Registro de Horas (Superado)
  - Comprobar que el diseño esté bien.
  - Comprobar la navegación y la interacción del calendario.
  - Comprobar que el botón de “hoy” muestra el día de hoy y que el botón “cambiar modo” conmuta entre la vista de mes y de semana.
  - Comprobar que muestra las imputaciones del día seleccionado y, en el caso de no haber, muestra un mensaje.
  - Comprobar que el botón “Añadir Registro de Hora” abre la vista de “añadir registro”.
  - Comprobar que al pulsar una imputación muestra la información detallada.
  - Comprobar que en la vista de información muestra bien los datos y permite borrar o editar la imputación en el caso de no haber ningún campo vacío.
  - Comprobar que el botón “Añadir” de la vista añadir registro funciona correctamente si están todos los campos rellenos.
  
- Correo (Superado)
  - Comprobar que el diseño esté bien.
  - Comprobar que el buzón de entrada y el de salida muestran un listado con los correos indicados.
  - Comprobar funcionamiento del Search Bar.
  - Comprobar que el botón del Navigation Bar de escribir un nuevo correo muestra la vista de “correo nuevo”.
  - Comprobar que si se pulsa un correo muestra la vista de “ver correo”.
  - Comprobar que al enviar un correo no haya ningún campo vacío.
  - Comprobar que al responder un correo no hay ningún campo vacío.
  - Comprobar que se borran bien los correos.

- Mi Perfil (Superado)
  - Comprobar que el diseño esté bien.
  - Comprobar que muestra la imagen y el nombre.
  - Comprobar que cada botón abre la vista apropiada.
  - Comprobar que guarda bien los datos de la edición de perfil.
  - Comprobar que cambia la contraseña.
  - Comprobar que hace logout.
  
- Recursos (Superado)
  - Comprobar que el diseño esté bien.
  - Comprobar que carga solo los grupos de recursos a los que el usuario tiene acceso.
  - Comprobar que carga bien los recursos de cada grupo.
  - Comprobar que muestra bien la información y que permite borrar un recurso o editarlo cuando están todos los campos rellenos.
  - Comprobar que añade bien un recurso.
  
- Empresas (Superado)
  - Comprobar que el diseño esté bien.
  - Comprobar que carga las empresas.
  - Comprobar que carga cada contacto de una empresa en concreto.
  - Comprobar que carga las diferentes aplicaciones de mapas con la ubicación correcta.
  - Comprobar que muestra la información de cada contacto y permite llamar o enviar un correo.
  
- Notificaciones (Superado)
  - Comprobar que las notificaciones funcionan correctamente.

Para el desarrollo de la aplicación se han hecho pruebas tanto en el simulador como en dos dispositivos diferentes, un iPhone 5 con iOS 8 y un iPhone 6s con iOS 9. De esta forma se podía comprobar que la aplicación funcionaba para ambas versiones, ya que existen ciertos métodos que están obsoletos y a veces no funcionan de una manera correcta. Además, al ser dos terminales con tamaños diferentes se podía comprobar que el diseño también era correcto en ambas pantallas. Además, mi supervisor realizaba una prueba al finalizar cada sprint, para asegurar que no escapaba ningún detalle.

Las pruebas de los servicios del backend se han realizado con Postman, un cliente de Google Chrome que permite realizar peticiones REST. Antes de integrar las llamadas en el código se comprobaba si los servicios devolvían los *json* correctamente.

## 5. Conclusions

After three months of development, the final application works and is stable. It has followed the methodology proposed and, as a result, the estimated deadlines have been met mostly by advancing the project much more enjoyable and with a feeling of almost total control.

One of the biggest advantages and one of the main reasons for the company to carry out this app is its versatility and standardization. This means that the fact of being translated eliminates the language barrier for internalization. Another important aspect is that it can be adapted to any company in any field, since the modules are common and changing the look with corporate colors or certain images would be very simple and fast.

When compared with similar applications in the market we see that the application combines the best of each and that there is a general interest in controlling companies in the world of mobile applications.

The fact that the side menu allows the user to navigate between the different modules is a point very much in favor, because it is like having many applications in one. It is an increasingly common trend and the different users are getting used to use it, therefore, is familiar.

I feel that the University has taught me a lot after carrying out this project. The fact that facing the development of an application in a programming language, unknown to me, almost made me realize the tools that I have been given by the different subjects and, in particular, programming courses. I believe that the research I have done would have been much more difficult without the prior training in those four years of degree.

Also, it is very interesting the independence I have gained since, from now on, I could program an application on my own and know where to find information or whom to ask in case a problem arises.

### 5.1 Future lines

This section deal with all those aspects in which the application can improve both functionality and modifications of current designs as the integration of new modules.



## 5.2 Design improvements

When users start using the application, suggestions of design modification will appear. For example, there may be buttons that would be more comfortable in another position of the screen or texts that are too small. Therefore, it is possible that in the future the application design could vary.

## 5.3 Code refactoring

The development has been conducted in the programming language Objective-c but Apple has launched a new one, Swift. It would be very interesting to consider in the future the development of the application in this new language that is now in its second version. It would not be surprising that many of the new methods were written exclusively in Swift and especially those related to Apple Watch, device that is expected a great evolution.

## 5.4 Expansion and modules integration

Since the beginning of the project the possibility of extending some of the modules of the application had been contemplated, but was rejected at first by the limited time we had. More modules were also proposed, but not fit within the initial estimate. These improvements and extensions are listed below.

In the timesheet module it is possible to add a timer to have a more accurate measure of the time an employee spend on each project.

The email notification functionality is not developed yet. What would do if were activated is to send an email to the user when the user receives a message or when charged one hour, among other actions.

At the beginning, the project was planned to integrate a module called Bill. This module would store information of different bills associated with the company and allows users with permission view them. As an extension, it was said the possibility that the application recognize the amount of the invoice from a photograph taken from the device, but it was too ostentatious raised.

While conducting the TFG, the responsible for the web application integrated new modules. These modules could also become part of the mobile application and are listed below.

The first module that was added is **My Documents**. In it, the user can see certain documents that are shared with it, such as payroll.

The second section that was added is **My Polls**. There are some surveys here that employees can fill. These surveys cover topics such as comfort in the workplace or where the Christmas dinner will take place.

The last module that was joined is **Material and Food**. It is an interesting module since it facilitates the work to the material staff. In it, the user can order office supplies or food.

## Bibliografía

- [1] Apple Inc., "About Objective-C", Apple Developer, Mac Developer Library, Programming with Objective-C, 2014 **[en línea]**. **Disponible en:** <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html> **[consulta:** 1 de junio de 2016]
- [2] Scrum, "An Empirical framework For Learning (Not a Methodology)", *Scrum Methodology*, Learn Scrum, 2015, **[en línea]**. **Disponible en:** <http://scrummethodology.com/> **[consulta:** 2 de junio de 2016]
- [3] Josema Santos, "7 tendencias en el desarrollo de aplicaciones móviles", *Aditiva Mobile Marketing*, 2016 **[en línea]**. **Disponible en:** <http://aditiva.es/7-tendencias-en-el-desarrollo-de-aplicaciones-moviles/> **[consulta:** 2 junio 2016]
- [4] Glen Stanford, "Company History", *apple-histoy*, history, 1996-2016, **[en línea]**. **Disponible en:** <http://www.apple-history.com/h1> **[consulta:** 3 junio 2016]
- [5] Glen Stanford, "iOS", *apple-histoy*, by family, iOS, 1996-2016, **[en línea]**. **Disponible en:** <http://www.apple-history.com/h1> **[consulta:** 6 junio 2016]
- [6] iTunes, Harvest. **[en línea]**. **Disponible en:** <https://itunes.apple.com/es/app/harvest-time-expense-tracker/id355395846?mt=8> **[consulta:** 3 de junio de 2016]
- [7] iTunes, Clockson. **[en línea]**. **Disponible en:** <https://itunes.apple.com/es/app/clockson/id872703970?mt=8> **[consulta:** 3 de junio de 2016]
- [8] iTunes, TimeSheet. **[en línea]**. **Disponible en:** <https://itunes.apple.com/es/app/tarjeta-tiempo-timesheet-gestiona/id445811752?mt=8> **[consulta:** 3 de junio de 2016]
- [9] iTunes, SugarCRM. **[en línea]**. **Disponible en:** <https://itunes.apple.com/es/app/sugarcrm-mobile/id599208633?mt=8> **[consulta:** 3 de junio de 2016]
- [10] iTunes, Noysi. **[en línea]**. **Disponible en:** <https://itunes.apple.com/es/app/noysi/id1071527690?mt=8> **[consulta:** 3 de junio de 2016]

- [11] iTunes, Mmmelon. **[en línea]. Disponible en:**  
<https://itunes.apple.com/es/app/mmmelon-comunicacion-y-trabajo/id1004330221?mt=8> **[consulta:** 3 de junio de 2016]
- [12] iTunes, MiniKeePass. **[en línea]. Disponible en:**  
<https://itunes.apple.com/es/app/minikeepass-secure-password/id451661808?mt=8> **[consulta:** 3 de junio de 2016]
- [13] ontsi, “Evolución del número de clientes de telefonía móvil en España”, 2016, **[en línea]. Disponible en:**  
<http://www.ontsi.red.es/ontsi/es/indicador/evoluci%C3%B3n-del-n%C3%BAmero-de-clientes-de-telefon%C3%ADa-m%C3%B3vil-en-espa%C3%B1a> **[consulta:** 4 de junio de 2016]
- [14] Fernando Rivero, “Informe ditrendia: Mobile en España y en el Mundo 2015”, ditrendia, 2015, **[en línea]. Disponible en:**  
<http://www.ditrendia.es/wp-content/uploads/2015/07/Ditrendia-Informe-Mobile-en-Espa%C3%B1a-y-en-el-Mundo-2015.pdf> **[consulta:** 4 de junio de 2016]
- [15] Dictamen 02/2013 sobre las aplicaciones de los dispositivos inteligentes, adoptado el 27 de febrero de 2013, **[en línea]. Disponible en:**  
[http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2013/wp202\\_es.pdf](http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2013/wp202_es.pdf) **[consulta:** 5 de junio de 2016]
- [16] Apple Inc., “Apple Push Notification Service”, 2016, **[en línea]. Disponible en:**  
<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html> **[consulta:** 7 de junio de 2016]
- [17] Apple Inc., “The Remote Notification Payload”, 2016, **[en línea]. Disponible en:**  
<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/TheNotificationPayload.html> **[consulta:** 7 de junio de 2016]
- [18] Google Developers, “Esquema de direcciones URL de Google Maps”, 2016, **[en línea]. Disponible en:**  
<https://developers.google.com/maps/documentation/ios-sdk/urlscheme> **[consulta:** 7 de junio de 2016]

## Glosario

- **Backend:** parte de la aplicación encargada de gestionar la información. Está oculta.
- **Wearable:** dispositivo que se lleva sobre, debajo o incluido en la ropa.
- **Notificaciones push:** notificaciones que tienen origen en el servidor.
- **App:** anglicismo que se utiliza para referirse a las aplicaciones móviles.
- **Sprints:** periodo de tiempo que engloba una serie de tareas de un proyecto y tras el cual se realiza una reunión.
- **Modelo-vista-controlador:** forma de organización del código.
- **Storyboard:** herramienta que ofrece *Xcode* para realizar el diseño de las vistas de una aplicación así como la navegación entre ellas.
- **Navigation Bar:** es la sección superior de la interfaz gráfica de una pantalla.

## Anexo I: Introducción traducida al castellano

### 1.1 Introducción del proyecto

En la actualidad, el mundo de las telecomunicaciones y, en concreto, el de la telefonía móvil está en constante desarrollo y las innovaciones técnicas están a la orden del día.

El uso de aplicaciones móviles y la adquisición de *smartphones* es cada vez más habitual en más sectores de la población y surgen nuevas necesidades que son cubiertas por estos terminales.

Por ello empecé a realizar mis prácticas de la Universidad en una empresa llamada *Armadillo Amarillo S.C.*<sup>21</sup>, dedicada al desarrollo de aplicaciones web y móviles, y durante ese tiempo aprendí a utilizar el lenguaje de programación *Objective-C* [1] de *Apple*. Al principio, me dediqué íntegramente al desarrollo de programas sencillos para familiarizarme con el entorno y con dicho lenguaje, pero en poco tiempo comencé a encargarme de ciertas tareas de proyectos reales.

Fue entonces cuando me propusieron hacer el Trabajo Fin de Grado con ellos, propuesta que acepté sin lugar a dudas. El Trabajo iba a ser una aplicación para PYMES llamada INTRANET.

El desarrollo se llevaría a cabo en lenguaje nativo, es decir, en el lenguaje de *Apple* anteriormente mencionado, *Objective-c*, e iba a ser la ampliación de un proyecto existente en la empresa. Este proyecto estaba formado por una aplicación web que realizaba una serie de labores de control interno de la empresa y de los empleados y un prototipo de aplicación móvil. La idea era que existieran dos aplicaciones para los dos sistemas operativos más frecuentes en el mercado: iOS y Android. Lógicamente, yo sería el encargado de la parte iOS y un compañero se encargaría de la parte Android.

Ambas aplicaciones constarían de los mismos módulos: uno para realizar imputaciones de las horas que le dedican los empleados a cada proyecto, denominado **Registro de Horas**; otro para enviar y recibir mensajes entre los miembros de la empresa, **Correo**; un tercer módulo sería el encargado de mostrar el perfil del empleado y permitir realizar ciertas modificaciones de sus datos personales, **Mi Perfil**; el módulo de **Recursos** mostraría los recursos a los que tiene acceso el usuario; y, por último, un módulo encargado de listar las empresas

---

<sup>21</sup> Armadillo Amarillo S.C., 2016, [en línea]. Disponible en: <http://www.armadilloamarillo.com/> [consulta: 1 de junio de 2016]

cliente, sus personas de contacto y la opción de abrir una aplicación de mapas para mostrar la ruta hacia dicha empresa, llamado **Empresas**.

El diseño de ambas aplicaciones lo realizamos nosotros mismos. Tanto mi compañero como yo (cada uno de su propia aplicación) nos encargamos de esbozar las diferentes pantallas que se iban a mostrar para que, más tarde, pudiéramos desarrollarlas, siempre teniendo cuidado de que ambas aplicaciones fueran lo más parecidas posible.

Las dos aplicaciones compartirían el mismo *backend* y la misma base de datos. El *backend* ya existía, puesto que era el que utilizaba la aplicación web, no obstante teníamos que ampliarlo y decidimos repartirnos, por módulos, el trabajo de implementación de los diferentes métodos que serían necesarios para que las aplicaciones funcionasen correctamente. Por su lado, la base de datos precisaba ligeros cambios, como añadir alguna columna a alguna tabla existente.

Por último, el proyecto incorporaría una aplicación *wearable*. Dicha aplicación gestionaría las notificaciones *push*.

## 1.2 Motivación

En la actualidad, cada vez más personas adquieren un *smartphone*, debido a ello que el mundo de las aplicaciones móviles está en pleno auge y la tendencia a su desarrollo es cada vez mayor.

No es de extrañar entonces que las empresas estén interesadas en digitalizar al máximo las labores cotidianas de las oficinas y que sea creciente la tendencia a utilizar terminales móviles para la mayoría de gestiones.

Es por ello que mi empresa me propuso realizar este proyecto. INTRANET es una aplicación que reúne las principales necesidades cotidianas de los trabajadores. Si echamos un vistazo al mercado, podemos encontrar ciertas *apps* cuyas funcionalidades son similares a alguno de los módulos citados en el punto anterior, pero ninguna los reúne. Por lo cual este proyecto puede ser de gran utilidad y le podría ser de interés a muchas empresas.

Personalmente, me considero una persona que intenta estar informada de las novedades técnicas que van surgiendo en diversos ámbitos tecnológicos pero, concretamente, siento gran interés por las innovaciones en tecnologías móviles. Es por ello que intento adquirir el último terminal lanzado al mercado (siempre que la economía me lo permite), descargar nuevas aplicaciones para probarlas, ver hasta dónde pueden llegar, técnicamente hablando, y descubrir bugs o posibles fallos que puedan tener.

Además, suelo tener ideas de aplicaciones que antes no podía llegar a desarrollar. Por ello me interesaba hacer el Trabajo de Fin de Grado de una aplicación móvil para una empresa. Es una manera de agradecer a *Armadillo Amarillo S.C.* lo que me han enseñado y, por otro lado, me ha permitido adquirir cierta autonomía y la posibilidad de darle forma a mis ideas.

Por último, hace 6 años llegó a mis manos un iPhone 4 y un año más tarde adquirí mi primer MacBook Pro. Desde entonces he sentido gran admiración por la marca de la manzana y aprender a desarrollar aplicaciones y poderlas probar desde mi propio terminal era una de las metas que me planteé desde que cursé las diferentes asignaturas de programación propuestas en mi Grado.

### 1.3 Alcance del proyecto

El proyecto se puede dividir en tres grandes bloques: aplicación móvil, aplicación wearable y *backend*.

- **Aplicación móvil**
  - Login: pantalla que aparece al iniciar la aplicación para introducir el nombre de usuario y la contraseña.
  - Panel Principal: pantalla que aparece al hacer login. Muestra un resumen de los tres últimos registros de horas, los tres últimos mensajes y los tres últimos recursos.
  - Registro de Horas: módulo de la aplicación donde puedes imputar las horas que le has dedicado a algún proyecto en concreto y consultar las imputaciones de algún día específico.
  - Correo: módulo de la aplicación que consiste en un sistema de mensajería para enviar y recibir mensajes entre los empleados de la empresa.
  - Mi Perfil: pantalla en la que puedes consultar y modificar tu perfil y activar o desactivar las notificaciones, entre otras funciones.
  - Recursos: módulo de la aplicación que consiste en consultar ciertos nombres de usuarios y contraseñas que pueden ser útiles para los empleados.



- Empresas: módulo de la aplicación en el cual puedes ver los clientes de la empresa, su persona o personas de contacto y lanzar una aplicación externa con la ubicación de dicho cliente.
- Menú Lateral: es un menú lateral desde el cual puedes acceder a todos los módulos y vistas nombrados.
- **Aplicación wearable**
  - Notificaciones: el Apple Watch mostrará una notificación cuando le llegue un nuevo mensaje al usuario y permitirá responderle con una serie de mensajes prefijados.
- **Backend**
  - Servicios: a través de servicios del *backend* conseguiremos una comunicación entre la aplicación y la base de datos.
  - APNS (Apple Push Notification Service): las notificaciones push se manejarán a través del *backend* con llamadas al servicio Apple Push Notification Service.

## 1.4 Metodología y calendario

Para el desarrollo del proyecto he seguido la metodología llevada a cabo por mi empresa, que consiste en una mezcla entre la metodología Scrum [2] y el reparto de tareas en *sprints*.

He utilizado un controlador de versiones llamado *SourceTree*<sup>22</sup> que permite ir guardando los cambios que se realizan en el código pudiendo retroceder a cualquier estado anterior del proyecto. Estos cambios son instantáneamente almacenados en local y en un servidor remoto, de tal forma que puedes acceder a tu proyecto desde cualquier lugar con el único requisito de contar con una conexión a internet.

Por otro lado, se lleva a cabo un reparto de tareas en *sprints*. Un *sprint* es un conjunto de tareas específicas del proyecto denominadas *historias de usuario*. Tras finalizar cada uno de los *sprints*, que en nuestro caso duran 2 semanas, se reúne el equipo para ver el avance y los posibles problemas que hayan podido surgir.

---

<sup>22</sup> Controlador de versiones *SourceTree*: <https://www.sourcetreeapp.com/>

El calendario que hemos seguido ha sido el siguiente:

- Sprint 0 del 1 de febrero hasta el 9 de febrero de 2016.
- Sprint 1 del 10 de febrero hasta el 24 de febrero de 2016.
- Sprint 2 del 25 de febrero hasta el 20 de marzo de 2016.
- Sprint 3 del 14 de marzo hasta el 27 de marzo de 2016.
- Sprint 4 del 28 de marzo hasta el 11 de abril de 2016.
- Sprint 5 del 12 de abril hasta el 26 de abril de 2016.
- Sprint 6 del 27 de abril hasta el 11 de mayo de 2016.

### 1.5 Fases del trabajo

Se ha dividido el desarrollo del proyecto en tres grandes fases: conceptualización, diseño y análisis; desarrollo y QA (Quality Assurance o pruebas de calidad); y redacción de memoria.

- **Conceptualización, diseño y análisis.**
  - Sprint 0
    - Conceptualizar la idea de la aplicación.
    - Diseñar todas las vistas de la aplicación.
- **Desarrollo y QA.**
  - Sprint 1
    - Diseño y funcionalidad del Login/Logout.
    - Diseño y funcionalidad del Menú Lateral.
    - Diseño del Panel Principal.
    - Implementación del Connection Manager.
  - Sprint 2
    - Diseño y funcionalidad de Registro de Horas.
    - Diseño y funcionalidad de Recursos.
    - Diseño y funcionalidad de Mi Perfil.

- Sprint 3
  - Funcionalidad del Panel Principal.
  - Diseño y funcionalidad de Correo.
- Sprint 4
  - Diseño y funcionalidad de Empresas.
  - Integración con APNS (Apple Push Notification Service).
- Sprint 5
  - Diseño y funcionalidad de Editar Foto de Perfil.
  - Gestión de Notificaciones.
  - Diseño y funcionalidad Apple Watch.
- **Redacción de la memoria.**
  - Sprint 6
    - Redacción de la memoria.

## 1.6 Medios empleados

Para el desarrollo de este proyecto he utilizado los siguientes medios:

- **Hardware:**
  - MacBook con 8Gb de memoria RAM y SSD de 250Gb
  - iMac 8Gb de memoria RAM y SSD de 250Gb
  - iPhone 5
  - iPhone 6s
  - Apple Watch de 38mm
- **Software**
  - Sistema operativo OS X El Capitan (versión 10.11.3)
  - Sistema operativo iOS 8.4
  - Sistema operativo iOS 9.3.1
  - Sistema operativo watchOS 2

## 1.7 Estructura del documento

He dividido la memoria en 5 capítulos: introducción, estado del arte, descripción y organización, desarrollo y conclusiones.

En el capítulo del **estado del arte** se realiza un análisis del entorno de desarrollo que se ha empleado, se hace un estudio de aplicaciones similares que están en el mercado y se evalúan las metodologías ágiles empleadas.

En el tercer capítulo, **descripción y organización**, se explica en primer lugar el estado del proyecto en el momento en que comienzo el desarrollo. También se muestra cómo se ha dividido el desarrollo y qué se ha hecho en cada parte, además de explicar la metodología que se ha seguido y las estimaciones que se hicieron.

En el capítulo de **desarrollo** se explica, módulo a módulo, cómo se ha programado cada funcionalidad a grandes rasgos. También se estudian ciertas alternativas al diseño y se muestra cómo se ha dividido el proyecto en directorios para facilitar su inteligibilidad y tenerlo todo ordenado. Al final se explican las pruebas a las que se ha sometido la aplicación para identificar fallos en la programación o bugs.

El último capítulo es el de las **conclusiones**. En él se explica cómo podría evolucionar la aplicación en un futuro, con ciertas modificaciones o con la ampliación y/o integración de nuevos módulos.

## Anexo II: Conclusiones traducidas al castellano

Después de tres meses de desarrollo, la aplicación final funciona y es estable. Se ha seguido la metodología propuesta y gracias a ello los plazos estimados se han cumplido en su mayoría, haciendo el avance del proyecto mucho más ameno y con una sensación de control casi total.

Una de las mayores ventajas y uno de los principales motivos de la empresa para llevar a cabo esta app es su versatilidad y estandarización. Esto quiere decir que el hecho de estar traducida elimina la barrera del idioma para su internalización. Otro aspecto importante es que puede adaptarse a cualquier empresa de cualquier ámbito, ya que los módulos son comunes y cambiar el aspecto con colores corporativos o imágenes determinados sería muy sencillo y rápido.

Si la comparamos con las aplicaciones del mercado similares vemos que reúne lo mejor de cada una y que existe un interés general por el control de empresas en el mundo de las aplicaciones móviles.

El hecho de que el menú lateral te permita navegar entre los diferentes módulos es un punto muy a favor, ya que es como tener muchas aplicaciones en una. Es una tendencia cada vez más extendida y a la cual se están acostumbrando los diferentes usuarios, por lo cual, les resulta familiar.

Siento que la Universidad me ha enseñado mucho después de llevar a cabo este proyecto. El hecho de enfrentarme al desarrollo de una aplicación en un lenguaje de programación prácticamente desconocido para mí me ha hecho darme cuenta de las herramientas que las diferentes asignaturas me han proporcionado y, en concreto, las asignaturas de programación. Considero que la labor de investigación que he llevado a cabo habría sido mucho más difícil sin el entrenamiento previo de estos cuatro años de Grado.

Es muy interesante también la independencia que he adquirido, puesto que, a partir de ahora, podría programar una aplicación por mi cuenta y sabría donde buscar información o a quien preguntar en el caso de que me surgiese un problema.

## 5.1 Líneas futuras

En este apartado se tratan todos aquellos aspectos en los que la aplicación puede mejorar, tanto modificaciones de las funcionalidades y diseños actuales como la integración de módulos nuevos.

## 5.2 Mejora de diseño

Cuando comiencen los usuarios a utilizar la aplicación irán apareciendo sugerencias de modificación de diseño. Por ejemplo, pueden existir botones que serían más cómodos en otra posición de la pantalla o textos que se ven demasiado pequeños. Por ello no se descarta que en un futuro el diseño de la aplicación pueda variar.

## 5.3 Refactorización del código

El desarrollo se ha llevado a cabo en el lenguaje de programación *Objective-c* pero Apple ha lanzado uno nuevo, *Swift*. Sería muy interesante plantearse en un futuro el desarrollo de la aplicación en este nuevo lenguaje que ya va por su segunda versión.

No sería de extrañar que muchos de los nuevos métodos estuviesen escritos exclusivamente en *Swift* y, sobre todo, los relacionados con el Apple Watch, dispositivo del que se espera mucho aún.

## 5.4 Ampliación e integración de módulos

Desde el comienzo del proyecto se había contemplado la posibilidad de ampliar alguno de los módulos de la aplicación, pero se desestimó en un principio por el tiempo limitado que teníamos. También se propusieron más módulos, pero tampoco cabían dentro de la estimación inicial. Dichas mejoras y ampliaciones son mencionadas a continuación.

En el módulo de Registro de Horas cabe la posibilidad de añadir un cronómetro para medir con mayor exactitud el tiempo que le dedicas a cada proyecto.

La funcionalidad de notificaciones de email no está desarrollada aún. Lo que haría si estuviese activada sería enviar un email al usuario cuando recibe un correo o cuando imputa una hora, entre otras acciones.

Al comienzo del proyecto estaba planeado integrar un módulo llamado Facturas. Éste módulo iba a almacenar información de las diferentes facturas asociadas a la empresa y las mostraría a los usuarios con permiso para verlas. Como ampliación, se planteó la posibilidad de que la aplicación reconociese el importe de la factura desde una fotografía tomada desde el dispositivo, pero era demasiado ostentoso.

Mientras realizaba el TFG, los encargados de la web integraron nuevos módulos a la aplicación web. Estos módulos podrían también pasar a ser parte de la aplicación móvil y son mencionados a continuación.

El primero de los módulos que se añadió fue el de **Mis Documentos**. En él puedes ver ciertos documentos que son compartidos contigo, como por ejemplo las nóminas.

La segunda sección que se añadió es la de **Mis Encuestas**. En ella se publican encuestas que los empleados pueden rellenar. Dichas encuestas tratan temas como la comodidad en el entorno laboral o temas ociosos como dónde se hará la cena de navidad.

El último módulo que se adhirió fue el de **Material y Alimentos**. Es un módulo interesante puesto que facilita el trabajo al personal encargado del material. En él puedes solicitar material de oficina que te falte o alimentos que eches en falta.

## Anexo III: Abstract

### Abstract

Nowadays, the world of telecommunications is constantly increasing because of the growing number of people that acquire *smartphones*. So there has emerged the need to create apps that meet people needs.

That's why I started my college practices in a company called *Armadillo Amarillo S.C.* There was a web application called **INTRANET** so they request me to do the iOS mobile and wearable application and request a partner to do the Android mobile and wearable application.

I would use the programming language called *Objective-C* of *Apple Inc.* and the software *Xcode* to compile the code.

For the development, the company gave me a MacBook because is the only way *Xcode* can be launched. Later, it will be replaced by an iMac.

To test the application, I used an iPhone 5 and an iPhone 6s like real terminals and the simulator to check other screen sizes and the different versions that the application will bear iOS 8 and iOS 9.

INTRANET is an application whose main function is to help companies to carry out with the business. It's simple to do if all the principal tasks are together in one app.

Doing a market study I found several apps whose functions were related with the time control, the client management, the communication and resources management, but no one had all of them.

Because of that, the mobile application is composed of several modules, such us **Principal Panel**, in which the last three time tracks, mails and resources are displayed; **Time Track**, where the user can see its time tracks in any day, add a new one or delete one existing; **Mail**, sending and receiving messages between the company members is allowed; **My Profile**, module where the user can find its personal information, modify it or change its password; **Resources**, that shows a list with the resources the user has access to; and the last module **Companies**, in which the user can see the clients its company is working with and open a maps app to see the route to their office.



The wearable application manages some push notifications like when the user receives a new message or when the notifications are activated.

The application uses a *backend* and a database. Both of them existed when I started the development but the *backend* didn't have all the methods that were necessary and there were some new fields needed in the database.

In order to further a development process as organized as possible I followed a methodology called *Scrum*. This methodology is based on three main points.

The first of them is to use a driver version called *SourceTree*. This software let a team to work in the same project simultaneously so it makes things going faster.

The second is to use *Jira*, a project manager that helps organizing tasks and have an overview of how the project progresses. This point introduces the third one, the distribution of tasks in *sprints*.

The tasks are called *user stories*. A *user story* is a specific task, such as making the design of one particular view. A *sprint* is a set of user stories that lasts two weeks. When a *sprint* finishes, the team makes a review, where the problems that may have arisen can be studied and where the starting signal is given to start the next *sprint*.

As to **project structure**, I've organized the code following the *MVC (Model View Controller) software architecture*. This is a method that stands up for the separation of concepts and code reuse for a more efficient and cleaner application, with the aim of facilitating the development and maintenance.

*Xcode* is organized in directories, like a computer, so, following the good practises, this directories organization have to agree with the organization of the directories of the computer.

The INTRANET project has two large folders, one for the mobile app and other for the wearable app.

At the same time, the mobile app folder has more directories for the images, *Storyboards*, models, helpers, managers, *ViewControllers* and supporting files.

The images are organized in a directory called *Assets.xcassets* that is the one to choose the right image size depending on the terminal that launches the application.

A Storyboard is an illustration of a view that helps the developer to make the design. The use of *constraints* make easier this work because thank to them, a design element can take place wherever the designer want and it doesn't matter what screen size will display it, it will always be on the same site.

The models are the responsible for storing the information of an object. They will have as arguments as columns have their table in the database. These are the models of the *MVC*.

Both helpers and managers are some help to do some functions that have to be done in different parts of the application, such us the connection with the *backend* or show alerts.

The ViewControllers are the controller part of the *MVC*. They are classes entrusted to do all the functionalities that the view has to perform. There is a ViewController for each view.

Supporting files are files such us the *AppDelegate*, responsible for the basic operation of the app or the *Localizable.strings*, responsible for the translation of the app.

According to the **application description**, the first view the user sees when the app launches is the **Login** screen, in which the user has to introduce the username and the password in order to enter the INTRANET. If the login works, the user will be shown the Principal Panel.

The **Principal Panel**, like previously mentioned, present the last three time tracks, mails and resources. If the user taps one of them, it will see its details. There are three buttons that allow it to add a new time track, send a message or add a new resource.

If the user taps the *Armadillo Amarillo's* icon, the **Lateral Menu** will appear, showing its profile image and its name just above a list with all the modules of the INTRANET.

In the **Time Track** module there is a calendar in which the user can see the time tracks for one particular day. If the user taps one, can see the details of a specific time track or delete it. This module also offers the possibility to add a new time track in the selected day.

The third module is **Mail**. When is tapped, the user can choose between see the inbox or outbox. With a button located on the *Navigation Bar* the user can send

a message to any employee of the company. It can also see a particular mail and reply to it or delete it.

**My Profile** is the module that has the personal information of the user. In it, it is possible to can change some information such us the name or the email address and change the password too. Here, the notifications can also be configured and

In the **Resources** screen a list of the groups of resources the user has access to is displayed. If one specific group is selected, the app will show another list with the resources of this particular group. Then, the resources can be edited, deleted of added.

The last module is **Companies**, in which the user can see a list of the clients the company is working with. If one client is tapped, the sales contact will be shown and the user can access to their personal information. There is the possibility to open a maps application to see the route to their office.

The **ConnectionManager** made the connection between the application and the *backend*. The requests are made by *POST*, it means that the application has to provide some arguments like the service that wants to be called; the auth, or token; and the args, that are the different arguments the service need to work.

When a request is done, the *backend* answer is a *json* with all the information demanded. Then, the application creates a model with this information and works with it to show it or to do whatever is needed.

The notifications are managed with **APNS (Apple Push Notification Service)**. This is the service that *Apple* offers to handle up with push notifications. APNS gives a device id to the device that is using the app and then, is the app that decides when a push notification has to be shown, what it says and what happens if the user taps the notification field.

Both GPS connection, making a phone call and sending an email uses third-party applications. The first one offers the possibility to choose between iOS Maps or Google Maps, the application just have to pass the location. To make a phone call, the INTRANET uses the iOS Phone Call app, just passes the phone number. Finally, to send an email, the application uses the iOS Mail app. In this case, the parameters that are passed are the email address of the receiver and a signature in the bottom of the message.

To make sure the application works well and there are no mistakes or bugs, there is a **QA (Quality Assurance)** test.

The QA consist of a series of tests that are made during the developing progress and after each *sprint*. The developer has to be testing every time that changes some in the application and other person, usually the PM (Project Manager), is the one who test the application when a *sprint* ends.

After three months working hard the application runs and is stable. Due to follow the proposed organization methodology the deadlines have been met.

The strengths of the application are versatility and standardization. That means that any company that wants an INTRANET could get it just changing the images and the corporate colours of the original app.

The idea of use a lateral menu is perfect because is very common in other applications and the users are used to see it.

But the application is not perfect and there are certain changes or additions that can be made.

For example, the module of Time Track could incorporate a timer in order to know the exactly time an employee engaged a project. The functionality of email notifications could be implemented too.

During the development, the people who manage the web application did some new modules that could be added to the mobile app too. These modules are **My Documents**, in which the user can see certain documents that are shared with it; **My Polls**, where there are some surveys that employees can fill; and **Material and Food**, an interesting module in which the user can order office supplies or food.

Another change could be the code refactoring by translating it into the new programming language of *Apple*, call *Swift*.